Distribution Agreement

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this dissertation. I retain all ownership rights to the copyright of the dissertation. I also retain the right to use in future works (such as articles or books) all or part of this dissertation.

_____ _____

Shrey Gupta Date

Transfer and Integration of Knowledge for Irregular Spatiotemporal Data

By

Shrey Gupta
Doctor of Philosophy

Department of Computer Science

Avani Wildani
Advisor

Andreas Züfle
Advisor

Yang Liu
Committee Member

Emily Wall
Committee Member

John Krumm
Committee Member

Accepted:

Kimberly Jacob Arriola, PhD
Dean of the James T. Laney School of Graduate Studies

December 13, 2024

Date

Transfer and Integration of Knowledge for Irregular Spatiotemporal Data

By

Shrey Gupta
M.Tech., Indraprastha Institute of Information Technology, Delhi, 2017
B.Tech., Guru Gobind Singh Indraprastha University, 2015

Advisors: Avani Wildani, Andreas Züfle

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in the Department of Computer Science
2024

Abstract

Transfer and Integration of Knowledge for Irregular Spatiotemporal Data
By Shrey Gupta

This dissertation aims to improve the predictive performance of transfer learning models on irregular spatiotemporal data. Modeling irregular spatiotemporal data collected via ground-sensors is complex due to spatial and temporal irregularities such as sparse observations and missing temporal points. Transfer learning on such data is much harder as it requires capturing the existing spatial/temporal irregularities, translating their dependencies across domains, as well as managing the distribution shifts during the transfer process. Therefore, this dissertation has three objectives: improve the generalizability of transfer learning models for regression (continuous-valued data), improve transfer across irregular spatiotemporal data sharing similar feature space, and improve transfer across irregular spatiotemporal data with dissimilar feature space.

We believe achieving these objectives will lead to solutions tailored to handle transfer for spatiotemporal data containing high-dimensionality, heterogeneity between domains, and spatial/temporal irregularities. An application that motivates the theme of this dissertation is pollution prediction for regions with few and sparsely distributed sensors as observed in many developing countries. Designing accurate prediction models for these regions is difficult due to existence of topographical, meteorological, geographical, and temporal dependencies. These dependencies have to be accounted for in the current *state-of-the-art* transfer models. Hence, the algorithmic improvements achieved in this dissertation would serve as a roadmap to apply and improve transfer learning for such domains.

Transfer and Integration of Knowledge for Irregular Spatiotemporal Data

By

Shrey Gupta
M.Tech., Indraprastha Institute of Information Technology, Delhi, 2017
B.Tech., Guru Gobind Singh Indraprastha University, 2015

Advisors: Avani Wildani, Andreas Züfle

A dissertation submitted to the Faculty of the
Emory College of Arts and Sciences of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Department of Computer Science
2024

Acknowledgments

"It takes a village to raise a PhD." translated from *"It takes a village to raise a child."*
"Social support is the oil that fuels PhD success and keeps it moving." – Dr. S. Cornér

Although the quotes above are valid for any doctoral journey, they resonate especially for mine, which would be impossible without the support of my dissertation committee and loved ones.

Firstly, I would like to thank *Dr. Avani Wildani* for giving me this opportunity to work on my doctorate under her supervision. Avani provided me this opportunity (i.e. a lifeline) when I was seeking an escape from a doctoral program that wasn't a good fit for me. I have learned a lot from her empathy and collegiality, seldom observed in high-performance environments. Avani has been a friend more than an advisor, defying conventional meritocratic norms. I am honored to have her as my advisor.

Dr. Yang Liu has been an anchor and a guiding light during this journey. He has been an advisor-adjacent; being there since the beginning; listening to my concerns, and dilemmas; providing me space to vent my frustrations and share my findings each week. Yang, I promise to carry forward your example and lead with your approach when given the opportunity.

Avani and Yang have empowered me to feel confident in myself and persevere through hardships. They gave me the freedom to explore my interests and carve a niche that may not have yielded immediate benefits but allowed me to grow towards a goal worth pursuing.

I would like to thank *Dr. Andreas Züfle* for his support during a challenging phase of my doctorate. I first met Andreas as an anxious student at a conference. His compassion was humbling, which was reflected in his protective and benevolent nature throughout my doctoral journey. He always finds time in his schedule to meet and brainstorm my thoughts. I am honored that he accepted to co-advise me.

Dr. Emily Wall was crucial in teaching me how to design robust experiments. I published my first lead-author paper with her. I always try to emulate her perseverance and I am grateful to her for providing me this experience.

Dr. John Krumm taught me to recognize potential in both others and myself. I met him at a conference and, nervously asked if he would join my committee. He graciously accepted, raising the standards of humility that he overwhelmingly posseses. Thank you John!

I am deeply grateful to my collaborators: *Dr. Jianzhao Bi*, *Yongbee Park*, *Dr. Suyash Gupta*, and *Dr. Alireza Karduni* for their invaluable insights and support across various projects. During my doctoral journey, I have been fortunate to be a part of three labs: SimbioSys, CAV, and the Emory Remote Sensing Group. I am thankful to the members of these labs for their expertise, kindness, and patience in addressing my questions. I also want to thank my friends, colleagues, and professors at Emory for their support throughout this journey.

Lastly, I would thank my partner, *Parul Sharma* as well as my family. This journey seems impossible without Parul. She has witnessed my struggles and has been my strength through them. My mom, *Manju Gupta* and my brother, *Suyash Gupta* have been my constant support system. *Mom*, thank you for sharing my fears everytime I ranted them in front of you. Finally, this dissertation is dedicated to my *dad*, who would be proud of this milestone, even though he is not here to see it.

Dad, this is in your loving memory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Transfer Learning

The concept of transfer learning stems from reusing knowledge across domains, such that concepts/knowledge of one task can be applied (reused) on another task [48]. For example, learning to ride a bike develops skills like balance, spatial awareness, and core strength, that can be transferred to other physical activities. Similarly, when learning a new musical instrument like the guitar, transferable skills such as hand coordination, rhythm, and music theory gained from playing another instrument like the piano can be utilized. In coding theory/programming, foundational knowledge grained by learning one programming language can be transferred to learning a new programming language. This intuition of knowledge transfer is particularly useful in the field of machine learning when there are less data samples i.e. limited data [222, 216, 165]. The core principle of transfer learning governs that many tasks share underlying structures, and predictors, and therefore, can be exploited to improve learning (model training) in a new, limited-data domain (target domain). Hence, by utilizing a model pre-trained on data-rich domains (also source domain: datasets with large samples) and having

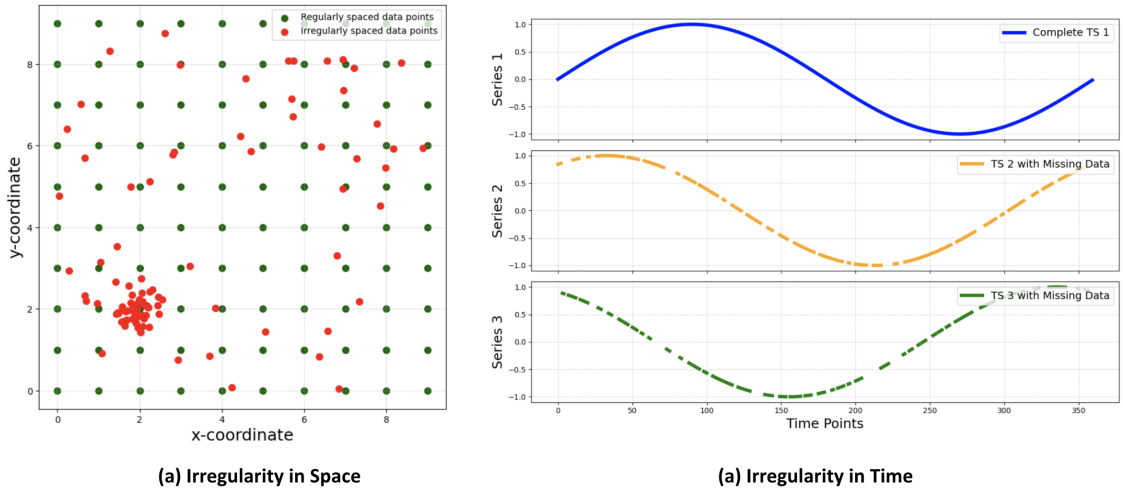

Figure 1.1: Irregularity in spatiotemporal data. (a) Shows irregularity is space – sparse data points (red) compared to grided data points (green). (b) Shows irregularity in time – three different time-series (TS) sampled over 360 time-points where TS1 is complete vs TS2 and TS3 have missing samples.

some similarity to the target domain, transfer learning allows for the reuse of knowledge. This reduces data collection needs (cost effective), time complexity (training time) and improves model prediction and generalization [13, 45, 222, 216, 165, 156, 257, 183].

Therefore, transfer learning can be defined as a **machine learning technique where a model trained on one task can be utilized to train a model on a new task with a smaller sample size and shared characteristics.**

Spatiotemporal Data and Irregularities

Spatiotemporal data consists of both space and time components, and hence, they capture how phenomena evolve over time and across certain locations. Such data is essential for analyzing domain trends such as tracking the spread of diseases over time to predict hotspots and formulate policies as part of epidemiological studies [115]. Similarly, traffic monitoring systems analyze congestion across road networks during the day to manage vehicle flow and reduce delays [130]. Even in the field of urban infrastructure development, the deployed smart sensors use IoT devices to monitor daily utility consumption and use the data to optimize utilization [180].

Irregularities in Spatiotemporal Data often occur in sensor dependent data collection use-cases such as environmental monitoring, traffic management systems and more [146, 2, 184]. While the spatiotemporal data is usually sampled synchronously (at fixed number of time intervals); irregular spatiotemporal data consists of asynchronous measurements collected at varying time points and spatial locations as shown in Figure 1.1. Although having irregular spatial locations is a common occurrence, but the varying time points (due to missing data) increase the difficulty of the machine learning prediction task. These irregularities limit the ability of the data processing methods to exploit the implicit temporal and spatial dependencies. Additionally, special framework is required in the learning model to address the complexities arising from these irregularities.

Environmental spatiotemporal data is characterized by the presence of spatial, temporal and spatiotemporal correlations (also dependencies) [181] and often suffer from similar spatiotemporal irregularities as mentioned previously. These datasets are a reflectance of topographical and meteorological variables interacting over space and time, influencing processes such as weather forecasting, wildfire behavior, air pollution levels, and other environmental phenomena. Capturing these interactions (or dependencies) is important, and hence, machine learning models offer a promising solution whereby they leverage convolutional and recurrent neural layers as one of multiple solutions for modeling spatial patterns and temporal sequences [194]. While the adeptness of machine learning models largely influences the prediction accuracy, however, the type of data collected is equally important — whether it is satellite observations [182], climate model outputs [108] or ground sensor measurements [7].

The satellite or remote-sensing data has coarser temporal resolutions with inaccuracies, and therefore are unable to capture local environmental variations. Additionally, their accuracy is often compromised due to atmospheric interference such as cloud cover or high surface reflectance) [181]. Whereas, ground sensors collect high-frequency

data at finer spatial resolutions, making them more accurate and suitable for capturing local environmental variations [7]. The collected data can be represented as grids, where measurements are taken at fixed location (generated by transforming the satellite or climate model outputs) [181], global/local images (satellite data) or spatiotemporal data obtained using ground sensor [7]. The satellite data has broad spatial coverage but lacks precision, whereas the ground sensor data captures localized variations but is often sparse and unevenly distributed.

Machine learning models for spatiotemporal data can be trained either on data from images or ground-sensor data. While the former provides spatially structured information, however, it captures limited complex and local variations [141]. Whereas, the latter creates data sparsity issues while training the ML models but consist of highly accurate, high-frequency data at finer spatial resolutions [246]. Hence, the spatiotemporal learning models address these challenges, by capturing existing dependencies within. For eg., graph-based neural networks or Gaussian processes are proficient in managing such sparse data distributions [227] as they consist of well-defined framework to capture such dependencies. Hence, the machine learning task is complex as the models requires balancing both the advantages (high accuracy) and limitations (irregular and sparse) of ground-sensor measurements, ensuring that the predictions are accurate.

The case for air pollution prediction also falls under the category of machine learning for spatiotemporal data. Air pollution contains pollutants such as PM2.5, PM10 (Particulate Matter i.e. particles with diameters up to 10 micrometers), ground-level ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2), and carbon monoxide (CO) [36]. They are caused due to vehicles, industries, and wildfires, and pose significant environmental and health risks. Among these pollutants, $PM_{2.5}$ with a diameter less than 2.5 micrometers, poses significantly higher risk as it is small enough to penetrate into the lungs or enter bloodstreams [131]. It is composed of a mixture

of harmful particles such as dust, soot, organic/inorganic chemicals, and heavy metals caused due to vehicle emissions, wildfires, and industrial processes [224] Hence, $PM_{2.5}$ can cause serious health risks such as respiratory diseases, cardiovascular issues, and premature death [191]. A similar case for utilizing ground sensors as opposed to satellite measurements can be made for collecting PM2.5 data – capturing more local variations allows for better model estimation and nuanced policy design.

The $PM_{2.5}$ ground-sensor problem

The $PM_{2.5}$ sensor problem is due to the scarcity of ground sensors as their installation is affected by cost limitations or resource deficiency. These ground sensors commonly have a region of cover of only a few kilometers [15, 99] and need to be densely installed. Hence, limitations such as complex topography of a region can cause difficultly deploying the sensors. Moreover, the cost of these sensors can range upto a few hundred dollars which might not be feasible for underfunded regions/countries [14, 99, 171].

The sensor problem can have multifold solutions such as (1) utilizing satellite observational data (if available) instead of installing ground sensors [204], (2) designing cost effective sensors useful for underfunded regions [253], (3) administrative intervention which involves government acquiring funds for new sensors [36], (4) designing prediction models independent or very less dependent on the data collection in the region [141]. Solution (1) which involves using satellite data has its advantages as it is widely available and has large range of land cover compared to the ground monitoring stations. However, it has lower accuracy as well as suffers from data collection issues due to optical variability such as cloudy weather or high surface reflectance [14]. Solution (2) has an advantage that they improve the monitoring coverage for developing regions/countries which have no regulatory monitoring framework as well as they can also be used to improve the fine-scale variability for PM2.5 by filling in the spatiotemporal gaps [15]. However, even these low-cost sensors cost around 2500 which is still

very high for many developing/underdeveloped countries. Moreover, the majority of low-cost sensors suffer uncertainty compared to reference grade sensors as they utilize the light scattering principle for measurement which is affected by variation in particle size, composition and shape [153]. Solution (3) which involves the government acquiring funds is a big ask/reach and may work for some regions/countries but might not work at all for other regions. Solution (4) is equipment independent where sophisticated machine learning models would be utilized to counter the dearth of data. We believe that transfer learning techniques fall in the category of Solution (4) and can be utilized to create prediction models for the PM2.5 domain as well as the broader use-cases involving irregular spatiotemporal fields [68]. Hence, understanding the transfer learning techniques catered towards irregular spatiotemporal data, analyzing their limitations and improving their performance is my goal and the underlying dissertation statement can be summarized as:

1.2 Dissertation Statement

This dissertation aims to design transfer learning models for irregular spatiotemporal data, which involves implementing algorithmic improvements for the current approaches and novel solutions to achieve a robust prediction performance.

1.3 Dissertation Objectives

Given the dissertation statement, I aim to answer the three research questions as shown in Figure 1.2 as well as listed below:

RQ1 **Robust and Generalizable Regression Transfer Learning**

What transfer approaches are suitable for regression problems since they closely relate to estimations for spatiotemporal (continuous) data. Are these approaches

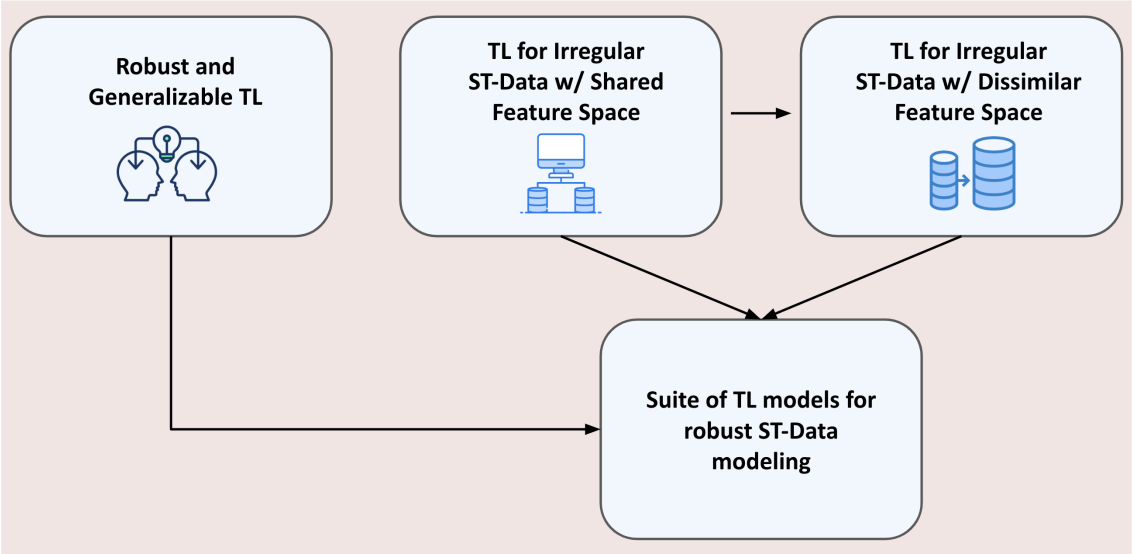

Figure 1.2: Dissertation Objectives

generalizable over multiple datasets with varying complexity?

We aim to evaluate the performance of various transfer learning methodologies for regression problems i.e. for continuous-valued data. Unlike classification tasks, regression requires precise estimation, making it important to utilize transfer learning models that can adapt to subtle variations present in the objective (target) domains. Our aim is to improve generalization for these transfer learning models by evaluating their performance across datasets of different complexities i.e. varying feature space, noise and sample size.

Additionally, while this objective aims to address limitations affecting the generalizability of regression transfer models, it also allows for deeper understanding into implementing reliable approaches for transfer for spatiotemporal data.

RQ2 **Transfer for irregular spatiotemporal data w/ shared feature space**

What patterns can be utilized to improve the prediction accuracy for the transfer across regions with similar feature space but spatiotemporal irregularities?

Transfer learning across regions consist of scenarios where the ground-sensors for

both the source and the target domains are irregularly and sparsely distributed with the source domain sensors being much larger than the target domain sensors. This uneven data availability affects estimation and poses a public health concern in developing countries with less available sensors. Additionally, both the source and the target domains share similar feature space and seasonality (consistent seasons around the year) such that there is no requirement for standardizing features across domains.

To mitigate the spatiotemporal irregularities, we identify transferable patterns that enhance the accuracy of transfer learning approaches. Our solution leverages spatial and semantic dependencies existing between the data-rich and data-poor regions to improve estimation in the latter using Instance transfer learning (ITL) models. Since, conventional ITL techniques struggle with complex dependencies such as topographical dependencies, meteorological dependencies, spatial and semantic autocorrelations present within and across domains, we identify such relationships and allow the transfer model to learn them for improved predictions.

RQ3 **Transfer for irregular spatiotemporal data w/ dissimilar feature space**

What patterns can be utilized to improve the prediction accuracy for transfer across spatiotemporal fields with dissimilar feature space, seasonality, temporal distinctness and sociocultural diversity (two geographically distant countries?

Transfer learning across regions with heterogeneous feature space is a complex tasks as it requires standardizing features between the domains. Our problem has an addtional difficulty since the source and target domains have differing seasonal cycle, sociocultural diversity and varying data distributions. This is usually the case with transfer learning between two geographically distant countries on opposite hemispheres. The spatiotemporal variability in addition to meteorological and environmental differences, creates difficulty to translate

the source domain knowledge.

The solution to this objective picks up from the previous one with improvements including data standardization, seasonality-agnostic experimentation across domain with annually-distinct data. Unlike traditional ITL techniques, our solution achieves improved transfer across regions by utilizing standardized feature set. Additionally, we also address the issues of cross-domain transfer challenges arising out of diverging meteorological, topographical, spatial/temporal patterns and feature spaces.

We believe that the suite of methodologies developed in this dissertation will be significant for spatiotemporal prediction modeling when the collected data is limited for a region. A central motivation behind this work has been to devise an executable solution for sensor-based data collection domains, such as air pollution prediction, where accurate and timely forecasting is crucial. Most regions, especially in developing countries, are bound by a small number of monitoring stations or limited environmental data. The framework of transfer learning proposed in this dissertation attempts to bridge this gap by knowledge transfer from data-rich to data-poor regions such that better predictions could be generated for under served areas. Additionally, these methodologies will help mitigate the challenges brought about by spatiotemporal data-irregularities, heterogeneity, and missing data-and hence are pertinent to domains for which the conventional ways of collecting data and making predictions do not work. This research provides a new perspective on how one might improve the accuracy of regional predictions based on variable data availability as well as embedding techniques that take into consideration spatial and temporal dependencies of environmental variables. These techniques will also be useful in the monitoring of public health and informing policy decisions for more responses to air pollution and other environmental hazards.

Chapter 2

Background and Related Works

Transfer learning involves transferring knowledge learned from one domain to predict labels for another domain. The knowledge is learned from an external domain called the source domain to train the prediction model and subsequently fine-tune it on the target domain [95]. While such a transfer learning is called supervised transfer as there exist a few target domain samples for fine-tuning, however, there also exist semi-supervised and unsupervised transfer whereby some unlabeled target domain samples or no target domain samples are utilized for model training and prediction. Transfer learning has multiple aliases and hence we can categorize them into two sub-classes: the first type of transfer models focus on sample size of the target domain, and hence, such a learning is known as *few-shot learning* (for limited target-domain sample: $\sim 1 - 100$ samples) [216, 197], *one-shot learning* (for 1 sample) [62], and *zero-shot learning* (for 0 samples) [228].

The second type of transfer models focus on the frequency of the source domain tasks, and hence, such a learning is known as meta-learning (i.e. learning to learn) or multi-task transfer learning [208, 248, 66]. Meta-learning involves the model learning from several diverse tasks which thereby allows it to learn new tasks more quickly and accurately through initialization of a generalized model and its weights [236]. It

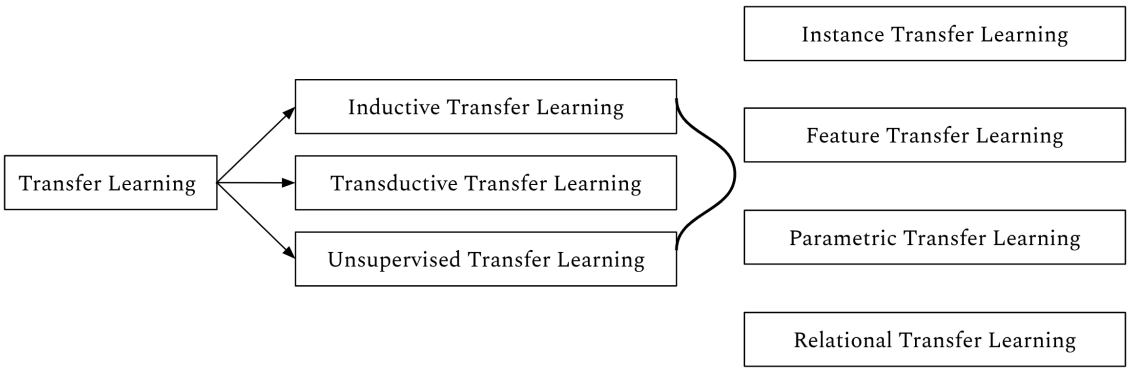

Figure 2.1: Categorization of Transfer Learning

has gained much popularity in the recent years as it allows for efficient sampling and generalization across the tasks. Multi-task transfer learning is rooted in multi-task learning where the goal is to learn multiple related tasks simultaneously, and generate shared representations that can be used for training the model. This can be extended to transfer learning where the multiple tasks are the diverse source datasets and the model can differentiate between the source and target learned representations [27]. Both meta-learning and multi-task learning are extensively used for modeling in domain such as computer vision, natural language processing, and robotics independently [241, 135].

Additionally, transfer learning can also be categorized into 3 categories based on source and target domain similarity and presence of labels [165, 222] as shown below.

1. **Inductive Transfer Learning**

   When the source and target domain are same whereas their tasks are different. For eg., both domains can be animal image classification, however, the source task is classifying animal whereas the target task is classifying the breed of an animal. It is represented as,

$$\mathcal{D}_S = \mathcal{D}_T, \quad \mathcal{T}_S \neq \mathcal{T}_T \tag{2.1}$$

2. **Transductive Transfer Learning**

When the source and target domain are different but their tasks are same. For eg., for the sentiment analysis task, the source domain is sentiment analysis for English language movie reviews and the target domain is sentiment analysis for Spanish language product review. It is represented as,

$$\mathcal{D}_S \neq \mathcal{D}_T, \quad \mathcal{T}_S = \mathcal{T}_T \tag{2.2}$$

3. **Unsupervised Transfer Learning**

   When both the source and target datasets are unlabeled and is represented as,

$$\mathcal{Y}_S = \mathcal{Y}_T = \emptyset \tag{2.3}$$

These categories can be further classified into 4 sub-categories, based on how the knowledge transfer process takes place, as *instance*, *feature*, *parameter*, and *relational* transfer. Hence, we first formulate the given conditions and subsequently discuss the 4 categories of transfer learning.

Formulation: Let $\mathcal{D}_S$ and $\mathcal{D}_T$ represent the source and target datasets, with inputs x_i, x_j and corresponding labels y_i, y_j. The prediction model f with parameters θ minimizes the error of the loss function $\mathcal{L}$. w_i represent weights for source samples based on their similarity to the target. Additional functions include ϕ for learning shared feature representations, and f_{map} to transfer relational structures $\mathcal{R}_S$ and $\mathcal{R}_T$ between domains. $\Delta\theta$, refine source parameters θ_S for use in the target domain, θ_T.

1. **Instance Transfer Learning**, involves using selective source domain samples. The selection is based on the similarity to the target-domain samples. It then combines the source and the 'few' target domain samples via domain adaptation and subsequently train a predictive model [26, 42, 232]. Hence, it can be represented by minimizing the combined loss over target domain and weighted

source domain samples.

$$\min_{\theta}(\sum_{x_i \in \mathcal{D}_S} w_i \mathcal{L}(f(x_i;\theta), y_i) + \sum_{x_j \in \mathcal{D}_T} \mathcal{L}(f(x_j;\theta), y_j)) \tag{2.4}$$

2. **Feature Transfer Learning**, involves standardizing the feature space or learning low-dimensional representations of the features which are shared between the source and target domain [5]. These learned representations are then used to train a transfer model. Hence, it can be represented by minimizing the loss over model f that learns low-dimensional or standardized representations of the source and target domains.

$$\min_{\theta,\phi} \sum_{x_i \in \mathcal{D}_S \cup \mathcal{D}_T} \mathcal{L}(f(\phi(x_i);\theta), y_i) \tag{2.5}$$

3. **Parameter Transfer Learning**, is also known as model transfer learning and contains learning based on sharing model parameters. Hence, a model is trained on the source domain samples and translated to the target domain by tuning the parameters/weights. [121, 18]. It can be represented as,

$$\theta_T = \theta_S + \Delta\theta \tag{2.6}$$

where $\Delta\theta$ is the adjustment learned on $\mathcal{D}_T$ to adapt θ_S for target tasks.

4. **Relational Transfer Learning**, focuses on data consisting of multiple dependencies/relations such as networked data (eg. social network data). The relationship between the data is transferred between the domains [151]. Hence, it can represented as,

$$\mathcal{R}_T = f_{\text{map}}(\mathcal{R}_S) \tag{2.7}$$

where f_{map} is a mapping function that transfers relational structures.

Since our focus is transfer learning for irregular spatiotemporal data, we primarily employ instance transfer learning (ITL) models as they are suited for continuous-valued (regression) domains. We also utilize parameter transfer models (i.e. neural models) as baselines for validation. While alternative approaches can be explored, the irregularity present in the spatiotemporal datasets make ITL models an optimal choice, as we observe in the later sections.

2.1 Instance Transfer Learning

Instance transfer learning (ITL) models are suited for domains with shared feature space as they combine the source and target domain information to achieve a successful transfer [45, 222, 165, 220, 143, 151, 199, 77, 161, 51, 257, 203]. The information is combined by adapting the source samples into *common structural representations* [100, 195, 38, 71, 195] that are subsequently combined with target samples to achieve transfer learning. ITL approaches are unaffected by missing data points as well as data sparsity, often present in spatiotemporal data making them ideal for such complex domains [37, 109, 30, 174, 148, 138]. Moreover, ITL methodologies are statistically interpretable [34] and accurate [220, 17], which increases their usability for domain experts [215] who avoid complex, black-box methodologies [11, 87, 106, 25, 178].

The current ITL methodologies can be vaguely divided into two categories based on how they apply the weighing strategy to the source domain instances. The first one involves re-weighing all the source instances at once using techniques such as *kernel mean matching* (KMM) [100, 38], *weighted-kernel ridge regression* [71], *Kullback-Leibler importance estimation* [195], translating samples to an invariant Hilbert-space [91], or learning sample weights based on the conditional distribution difference [29]. The second type of ITL category is the ensemble learning models that primarily include

the boosting methodologies [40, 169, 212]. They are useful in iteratively adjusting the weights as well as penalizing the instances that negatively affect the target learner. For the **RQ1** (chapter 3), we build an ensemble approach based on the boosting algorithm for transfer learning. Whereas, for the **RQ2**, we present a generalized improvement that can be utilized by both categories of ITL approaches.

Additionally, we formalize the problem of instance transfer learning (ITL) where $\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$ represent the labeled source domain and $\mathcal{D}_T = \{(x_j^T, y_j^T)\}_{j=1}^{n_T}$ represent the labeled target domain such that x_i is the feature space and y_i are the corresponding labels. ITL models involve reweighing the source domain samples where w_i are the weights assigned to each source sample, such that highly similar (to target samples) source samples are assigned a greater weight. These weights are then utilized into a modified loss function that adapts the source samples similar to the target samples as shown below.

$$\mathcal{L}_{\text{ITL}} = \sum_{i=1}^{n_S} w_i \cdot \ell(h(x_i^S), y_i^S) + \sum_{j=1}^{n_T} \ell(h(x_j^T), y_j^T) \tag{2.8}$$

where $\ell(\cdot)$ is a loss function (e.g., squared error for regression problems), $h(\cdot)$ is the hypothesis/predictive model, and w_i are weights for source samples based on their similarity to the target samples.

Instance Transfer Learning for Regression

The instance transfer learning for regression problems involve minimizing the discrepancy between the source and target distributions. For eg., in Kernel Mean Matching (KMM) ITL model [100], the weights w_i are optimized to reduce the difference in feature means between the two domains such that,

$$\min_{w} \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} w_i \phi(x_i^S) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\|^2 \tag{2.9}$$

subject to $0 \leq w_i \leq B$ for some bound B and $\sum_{i=1}^{n_S} w_i = n_S$, where $\phi(\cdot)$ is a feature mapping function. Since these weights are adjustable, the model is able to learn from weighted source samples closer in distribution to the target samples. These learned weights w_i are then utilized in a regression model, where the total loss across the combined (weighted source and target) data is minimized as,

$$\mathcal{L}_{\text{regression}} = \sum_{i=1}^{n_S} w_i \cdot (h(x_i^S) - y_i^S)^2 + \sum_{j=1}^{n_T} (h(x_j^T) - y_j^T)^2 \tag{2.10}$$

Hence, the KMM-ITL model allows to prioritize source samples most similar to the target samples, and thereby improves the generalizability of the model.

2.2 Machine Learning for Spatiotemporal Data

2.2.1 Spatiotemporal Data

Spatiotemporal data consists of measurements referenced in both space and time. It merges the spatial components with the temporal dimension creating a corpus for phenomena varying over time and across multiple locations [64, 259]. Similarly *environmental spatiotemporal data* consists of complex spatial, temporal, and spatiotemporal correlations that need to be efficiently captured for accurate modeling and forecasting [2]. These dependencies are especially prevalent in dynamical environmental domains such as air pollution, precipitation, wildfires and more [181].

Spatiotemporal data can be broadly categorized into two types based on how they are collected i.e. data collected from moving objects and data collected from stationary object [155]. Moving object spatiotemporal data captures dynamic trajectories of entities such as vehicles, animals, or people to track their space-time trajectories [70, 254, 263, 3]. Domains such as traffic analysis, wildlife tracking, and urban planning, that capture trajectories of moving objects consists of finding patterns or forecasting

entity locations [152, 4, 117]. For eg., vehicle trajectory data would record its varying location across a city, and would contain both spatial and temporal dimensions. This can be utilized in real-time traffic management or navigation systems [134].

Conversely, stationary object spatiotemporal data is collected via fixed sensors often used to record environmental or atmospheric variables over time in a location [2]. Such stationary ground-sensors record continuous measurements that are highly accurate but also variable as they are affected by surrounding parameters such as temperature, pressure, and more. Hence, modeling sensor data has its own unique challenges as the data is irregularly spread both spatially and temporally. The spatial irregularities can be caused due to imbalance in the frequency of installed sensors such that some areas witness high sensor density compared to others due to factors such as population, regulations, funding and more [192]. Similarly, temporal irregularities stem from inconsistent sensors monitoring where certain intervals are missed entirely due to diurnal sensor activity [193].

Satellite data, while now intrinsically spatiotemporal, can be transformed into spatiotemporal data by processing consecutive images over time [74, 61, 258]. Similarly, the temporal changes observed in the image data at avrying spatial scales can also be transformed into spatiotemporal data [124]. For eg., temporal scenarios such as monitoring air quality or the rate of deforestation [89] via satellite images can be converted into spatiotemporal measurements by converting these images into a series of raster grids. Yet satellite data have their own range of drawbacks, such as missing data points due to cloud cover [120], thereby reducing precision. Hence, satellite data is often useful for modeling high-level regional and global trends for a phenomena [181].

2.2.2 Prediction Models for Spatiotemporal Data

Deep learning and *state-of-the-art* machine learning models hold promise to address the challenges from spatiotemporal data, given their ability to extract patterns from

high-dimensional and complex datasets [129]. For image data, Convolutional Neural Networks (CNNs) are effective in learning localized patterns [122]. Whereas sequential models like Recurrent Neural Networks (RNNs) and their variants such as LSTM, transformers, and more, are effective in capturing concurrent dependencies [206].

Environmental Deep learning and machine learning models [181, 168, 226] are widely utilized for tasks like precipitation modeling, extreme weather prediction, and wind forecasting, as they are adept in capturing short-term (hours to weeks) [76, 107, 176, 177] and long-term dependencies [101, 8, 85]. For eg., spatiotemporal RNNs are effective in forecasting PM2.5 values using historical data with the locations spread regularly across space. However, such vision-based deep learning models have to be adapted to handle the complex environmental data collected using ground sensor and consisting irregularities. Unlike satellite data, which is often represented as spatiotemporal grids, sensor data often require sophisticated techniques to capture interactions across space and time.

2.2.3 Machine Learning for Pollution Prediction

Air pollution consists of harmful aerosols like Particulate Matter (PM), gases like CO_2, and volatile organic compounds; water through industrial discharges, agricultural runoff, waste, and pesticides,as well as heavy metals, and industrial wastes [123]. Hence, estimating air pollution is significant for safer public health and preserving ecosystems [24]. These pollutants can cause severe respiratory problems, can contaminate drinking water, degrade soil quality, and contributes to climate change [119]. While machine learning based spatiotemporal modeling has garnered high interest due to rise in deep learning, it also useful due to its success in designing models that can estimate pollutants such $PM_{2.5}$, PM_{10}, NO_2, O_3, and SO_2 [111, 101, 14, 15, 28, 230, 181, 162, 231]. These models are able to capture complex nonlinear dependencies existing between features (meteorological, topographical, and geographical) that also affects air quality.

In the case of machine learning models, Random Forest (RF), Support Vector Machines (SVM), and Gradient Boosting Trees (GBT) have been widely employed as they can successfully handle high-dimensional data and capture intricate patterns [244, 75, 217]. In the case of deep learning models, techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models are highly adept at extracting both spatial and temporal dependencies existing in the data [126, 57, 59, 213]. For example, Qi et al. developed a CNN-LSTM hybrid model for multi-pollutant prediction that learns spatiotemporal information from big data for a generalized training, and thereby successfully outperforms baseline prediction models [174]. There have also been hybrid models that fuse physical laws and equations with machine learning models to capture pollution dynamics [10].

Recent studies employ hybrid spatiotemporal machine learning models whereby they employ both deep learning and machine learning solutions for a comprehensive estimation. For example, Betancourt et al. [9] employ gradient-boosted tree (GBT) and multi-layer perceptron (MLP) algorithms to model near-surface nitrogen dioxide (NO2) and ozone (O3) concentrations at a high spatial and temporal resolutions, integrating satellite data with ground-level environmental and meteorological data. Additionally, Wang et al. [31] devise an innovative graph neural network (GNN) approach to capture spatial dependencies among monitoring stations as edges and nodes to forecast multiple pollutants and improve performance. These advanced modeling techniques, coupled with the integration of diverse data sources such as satellite observations, and ground-level data have a three-fold effect of improving the accuracy of pollution prediction [229, 98], understanding of pollution dynamics [184] and supporting public health interventions [190].

2.2.4 Machine Learning for $PM_{2.5}$ Prediction

PM2.5 particles are extremely harmful due to their small size, allowing them to penetrate deep into the lungs and even enter the bloodstream, causing respiratory and cardiovascular issues[172]. Accurate estimation of PM2.5 exposure is crucial for epidemiological studies and public health interventions, as it enables researchers to assess health risks and develop targeted pollution reduction strategies[21]. Machine learning (ML) and deep learning models have emerged as powerful tools for PM2.5 estimation and forecasting, addressing the limitations of traditional monitoring methods. Regression-based approaches, such as Random Forest (RF), Support Vector Regression (SVR), and Gradient Boosting Machines (GBM), have shown promising results in estimating PM2.5 concentrations from satellite data and other environmental variables [150, 221]. These models can capture complex non-linear relationships and handle high-dimensional feature spaces effectively. More advanced techniques like Geographically Weighted Regression (GWR) and Bayesian hierarchical models have been employed to account for spatial heterogeneity in PM2.5 distributions [132, 127, 16, 72]. Ensemble methods, combining multiple ML algorithms, have also demonstrated improved performance in PM2.5 estimation [150, 221].

Deep learning models have further pushed the boundaries of PM2.5 estimation and forecasting. Convolutional Neural Networks (CNNs) have been successfully applied to extract spatial features from satellite imagery and meteorological data for PM2.5 prediction [132, 127]. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have shown excellent capabilities in capturing temporal dependencies in PM2.5 time series data [150, 221]. Hybrid models, such as CNN-LSTM architectures, have been developed to leverage both spatial and temporal information for more accurate PM2.5 forecasting [132]. Advanced techniques like Graph Neural Networks (GNNs) and Transformer models have also been explored to model complex spatio-temporal relationships in PM2.5 data [127, 150]. Transfer

learning approaches have been successfully employed to adapt models trained on data-rich regions to areas with limited monitoring data, addressing the challenge of data scarcity in many parts of the world [221].

The measurement of ambient $PM_{2.5}$ exposure mainly relies on ground monitoring stations [83]. However, even in well-endowed countries/regions such as the United States, more than 70% of the counties do not consist of standardized $PM_{2.5}$ monitoring stations [243]. Moreover, this scarcity of ground monitoring stations is more prevalent in underdeveloped/developing countries [219, 147]. Hence, the last decade has seen the utilization of satellite-based remote sensing methodology [14, 99] and low-cost sensors [15] to extend the coverage for detecting the $PM_{2.5}$ levels. But, the satellite-based data has lower accuracy and suffers from data collection issues due to optical variabilities such as cloudy weather or high surface reflectance [14]. Transfer learning, which leverages models/data from another domain does not have limitations possessed by the above-mentioned approaches such as accessibility constraints for proprietary sensors or data engineering bottlenecks for satellite data, thus, making it optimal to generate prediction models using data collected from very few monitoring stations.

2.3 Transfer Learning for Spatiotemporal Data

Transfer learning models for spatiotemporal domain address the data collection challenges such as limited labeled data, out-of-distribution test data, and unsupervised learning i.e. unlabeled data [239, 251]. Convolutional Long Short-Term Memory (ConvLSTM) networks have shown promise in capturing both spatial and temporal dependencies such that the data can be translated into a series of grids and the attention mechanism of the model focuses on relevant features [32]. Adversarial training techniques have been used to align source and target domain distributions by generating learned representation that allows model to differentiate between the two domains,

thereby improving transferability [93]. Moreover, some meta-learning methods have been developed for quick adaptation to new tasks with limited data, an important aspect in spatiotemporal transfer learning [197]. Ma et al. [139] apply bi-directional LSTM where transfer learning is used to transfer the knowledge learned from smaller temporal resolutions to larger temporal resolutions (local temporal transfer learning) whereas Yao et al. [237] propose a differentiable framework called transferable memory that distills knowledge from RNN units and applies it to the target data using a novel structure called Transferable Memory Unit (TMU).

Transfer learning has also been successfully used in application for spatiotemporal data domain such as climate modeling, urban computing, and video analysis [85, 218]. Graph neural networks have been used to model the complex relationships existing whereby the generated graph is homogeneous or heterogeneous based on the task [225]. In the field of computer vision, self-supervised pretraining on large-scale video datasets has shown to be particularly effective for downstream spatiotemporal tasks [88, 256]. Recent work has also explored the use of contrastive learning techniques to learn transferable representations for an unsupervised learning i.e. unlabeled data [35, 175]. As this field evolves, there is a focus on more robust and general transfer learning models that can deal with the unique challenges related to spatiotemporal data; examples include temporal resolution variations or spatial heterogeneity [179, 251].

However, these solutions consider image/video/grid data for their experiments and cannot be utilized for continuous-valued datasets.

2.3.1 Spatial/Temporal Transfer Learning

Spatial transfer learning involves transferring a model learned over a particular space to a new space. Ferris et al. [65] utilized the Gaussian Process Latent Variable model to create a mapping function between the two domains. However, this model was complex as it required a pre-requisite motion-dynamics model. Pan et al. [164] introduced a

manifold regularization model. The challenge with such a model was to learn the labels of locations of a small part of the building when the model had been trained for a much larger area. Pan et al. [166] apply spatial transfer learning for the Wi-Fi localization data. However, the two spaces were within a building and thus cannot be considered highly divergent domains. In our objective (**RQ2**), we apply spatial transfer learning for two highly distant regions spread over a country.

Popular domain adaptation/transfer learning techniques for time series domain focus on leveraging domain-invariant and domain-specific representations of the data [173, 223, 73, 19]. However, these methodologies, designed without considering the sequential nature of time series data, are not readily applicable to forecasting tasks. Cai et al. [128] address the domain shift challenges by devising an approach that involves minimizing the disparity in the associative structure of the time series for the target and source datasets. However, such an approach is not suitable for multi-horizon forecasting tasks as its labels are associated with inputs rather than being pre-defined. Similarly, Hu et al. [97] introduced DATSING, that employs adversarial training to fine-tune pre-trained forecasting models. This fine-tuning process is achieved by augmenting the target dataset with selected source data based on predefined metrics. Although DATSING is an effective approach, its complexity due to having two distinct stages makes it cumbersome to use. Moreover, its solution does not incorporate domain-specific features for forecasting and thereby cannot be utilized for specific applications such as air pollution estimation. Jine et al. [105] utilizes a novel attention-sharing mechanism to generate domain invariant representations for the time series. However, these representations are generated for an unsupervised learning task as compared to our supervised learning task with few labeled target sample. Our proposed solution in **RQ3** hopes to improve upon this methodology and has been explained in detail in Chapter 5.

2.3.2 $PM_{2.5}$ Estimation via Transfer Learning

There has been some work in the space of $PM_{2.5}$ prediction which involves the usage of deep neural networks apt for time-series prediction. This involves the presence of consistent temporal points in the dataset and while a few studies impute data prior to the training, such an approach is ineffective for our case due to a large number of missing temporal points ($\sim 365k - 250k$). Ma et al. [140] present a stacked bidirectional LSTM transfer network that required consistent temporal samples and utilized methodologies such as rolling window [22] to impute data. Similarly, Fong et al. [67] also utilize data imputation methodologies to generate missing data and consequently create a transfer learning methodology combining LSTM and RNN for a spatiotemporal prediction. Fang et al. [60] proposed a hybrid strategy based on LSTM and domain adversarial neural networks (DANN), and similarly Ni et al. [159] presented a hybrid transfer model that utilizes Maximum Mean Discrepancy (MMD) for importance sampling of the source-domain samples and a two-phase model for feature transfer learning.

In addition to the problem of missing temporal points for the air pollution domain dataset, the prediction of $PM_{2.5}$ takes place over a large number of unknown testing sites by training on very few training sites (unlike the above studies where the train-test ratio was 70:30). Moreover, the testing sites are sparsely located over a large area (multiple states) making it highly challenging to apply the above transfer learning solutions. Hence, in **RQ2** and **RQ3** we present solutions that tackle the spatial and temporal transfer learning problem one at a time. The final goal involves combining these solutions for a "go-to" single methodology.

Chapter 3

Robust & Generalizable Regression Transfer Learning

This objective consists of experimental evaluation of the instance transfer learning (ITL) techniques suited for the regression domain where the source and target domains share the same feature space. During our experimental evaluation, we noticed that many ITL techniques performed sporadically i.e. imbalanced accuracies such that the models were highly accurate for certain datasets and inaccurate for other datasets. These models were tested upon a diverse set of datasets with varying complexities (also measuring the complexity of the dataset).

Hence, this objective focuses on presenting algorithmic improvements for a boosting based instance transfer learning methodology, TrAdABoost.R2 [169] to introduce a new generalizable and robust methodology, STrAdaBoost.R2 that performs consistently well over all experimental datasets.

This research objective **(RQ1)** was published in the journal – International Journal of Data Science and Analytics [79].

3.1 Rationale:

While semi-supervised and unsupervised learning methodologies work well for partially labeled or unlabelled datasets [163, 13], they fall short for instances where the sample size is small [216, 222, 165, 220, 66]. Instance-transfer learning (ITL) [222, 165, 220, 45, 77, 161, 51], a sub-class of transfer learning approaches [257], is designed for domains with limited and labeled samples, shared feature-space, and independent and identically distributed (i.i.d) data-distributions [167, 203], making it ideal for real-world datasets [37, 109, 30, 148, 138]. It stands apart from its counterparts, such as feature-transfer learning and parameter-transfer learning, as it allows data adjustment and transformation of domain instances, making it ideal for dissimilarly distributed source and target domains. Moreover, ITL methodologies are as statistically interpretable [34] as they are powerful [220, 17], which increases their usability for domain experts [215] who avoid complex, black-box methodologies [11, 87, 106]. Therefore, these methodologies have the advantage of being less complex but equally reliable when compared to deep learning based transfer methodologies. Another reason for leaning towards ITL methodologies is because it is easier to transfer the source domain by applying adaptation methodologies [100, 195] as well as using techniques involving reduction of distribution difference between the source and the target domain [38, 71, 195]. The accuracy of prediction does not just depend on the transfer learning methodology but also involves the nature of the distribution. Real-world datasets suffer from collecting data that is complete, high-resolution, and evenly sampled. This is due to the dependence on the cost of equipment which can result in hardware limitations. This leads to the resulting dataset varying in resolution as well as the quality [138]. Hence, a robust transfer learning methodology should perform consistently well for data distributions with varying complexities.

Among the ITL methodologies, we employ ensemble methodology, especially the boosting methodology [34] as it aggregates the results from multiple learners.

Similarly, the transfer boosting methodology TrAdaBoost.R2 [169] is regularized and uses domain adaptation for iteratively re-weighing the source instances with respect to the target dataset for knowledge transfer [201]. The underlying architecture is AdaBoost [188], which focuses on misclassified training instances, leading to contextual learning. However, boosting methodologies suffer from negative transfer [183] when the source dataset size is large compared to the target dataset, leading to a skewed final model. To address the problem of negative transfer, we introduce S-TrAdaBoost.R2, a successor to two-stage TrAdaBoost.R2 (TTR2) that uses importance sampling [165, 110, 250] to improve the alignment of source instances with the target values, and applies a balanced weight update strategy to mitigate the skewness generated due to the large sample size of source datasets. We test S-TrAdaBoost.R2 across a range of standard regression datasets with limited target instances and varying complexities, and find that it outperforms other ITL methodologies 63% of the times and the baseline TTR2 more than 75% of the times. Notably, it has consistent performance (RMSE and R-squared score) for both the regular comparative study and the Ablation study (Fig. 3.2 and Table 3.2), as opposed to fluctuating results as observed for other instance transfer methodologies. The primary contributions of this objective are:

1. We introduce S-TrAdaBoost.R2, complexity-tolerant, domain-agnostic boosting-based transfer learning algorithm that uses importance sampling and a balanced weight update strategy to outperform its predecessor TTR2 and other competitive ITL methodologies.

2. We discuss the complexity measures, i.e. metrics to quantify the complexity of distribution. They categorize the distribution based on correlation, linearity, and smoothness, to provide a numerical estimate of its simplicity.

3. We demonstrate that S-TrAdaBoost.R2 outperforms competitive ITL methodologies when measured in terms of accuracy and loss, for high-complexity datasets.

We also provide the ablation analysis for Importance Sampling, which demonstrates the modularity and commutability of the technique.

Hence, with the goal of designing, a robust transfer learning methodology that performs consistently well for data distributions with varying complexities, our model, STrAdaBoost.R2, is a complexity-tolerant, and domain-agnostic transfer algorithm that uses importance sampling and a balanced weight update strategy to outperform its predecessor, TrAdaBoost.R2 and other competitive ITL methodologies.

3.2 Background

Previous work on transfer learning [45, 222] provides methodologies for measuring the shared information content between multiple domains in transfer learning [143, 151, 199]. These models attempt to find common structural representations of source instances to gauge the quantity as well as the quality of the transfer. However, for highly dissimilar source and target domain instances, a reduction of prediction accuracy for transfer learning algorithms when compared to non-transfer learning algorithms i.e. *negative transfer* is commonplace [183]. Figure 3.1 shows negative transfer when TTR2 and AdaBoost.R2 are fitted over the concrete dataset from UCI machine learning repository [6]. We observe a decline in TTR2's performance as the target sample size increases. This shows a trade-off in the performance of transfer learning algorithms to the sample size of the target distribution. Hence, transfer learning algorithms perform better when the sample size of a target dataset is small.

The concept of translating knowledge and model across domains has been much researched upon and hence, transfer learning, similar to machine learning, is observed for both classical transfer learning [26, 38, 71, 169] and deep transfer learning methodologies [11, 87, 106, 139, 200, 235, 245, 264]. While deep networks can often improve

transfer accuracy, they sacrifice model interpretability, generalizability, adaptability, and flexibility for more diverse tasks [25, 178]. Whereas, ITL algorithms, unlike deep transfer models, do not suffer from obscurity in showing intermediary steps and learned concepts in order to have greater transparency. Even for unrelated source and target domains, the source instances adapt to the target instances by either re-weighting [26, 71] or transforming to the target space [38], indicative of the adaptability of ITL methodologies. The current ITL methodologies can be vaguely divided into two types based on how they apply the weighing strategy to the source domain instances. The first one involves re-weighing all the source instances at once using techniques such as Kernel Mean Matching (KMM) [100, 38], Weighted-Kernel Ridge Regression [71], Kullback-Leibler Importance Estimation [195], translating training instances to an Invariant Hilbert Space [91], or learning source domain instance weights based on the conditional distribution difference from the target domain [29]. The second type of methodology is the ensemble learning methodology, primarily including boosting techniques.

3.2.1 Boosting

Boosting [188] is an ensemble technique that builds a classifier by using a set of weak learners, whereby the weights of the training samples are updated over a chosen number of iterations, and finally these weak learners are combined to generate a strong learner. Popular boosting methodologies such as AdaBoost.R2 [50] typically assume that the test and training datasets have a similar distribution and hence do not require domain adaptation. They do not suffer from overfitting [198] and have a robust prediction over diverse datasets.

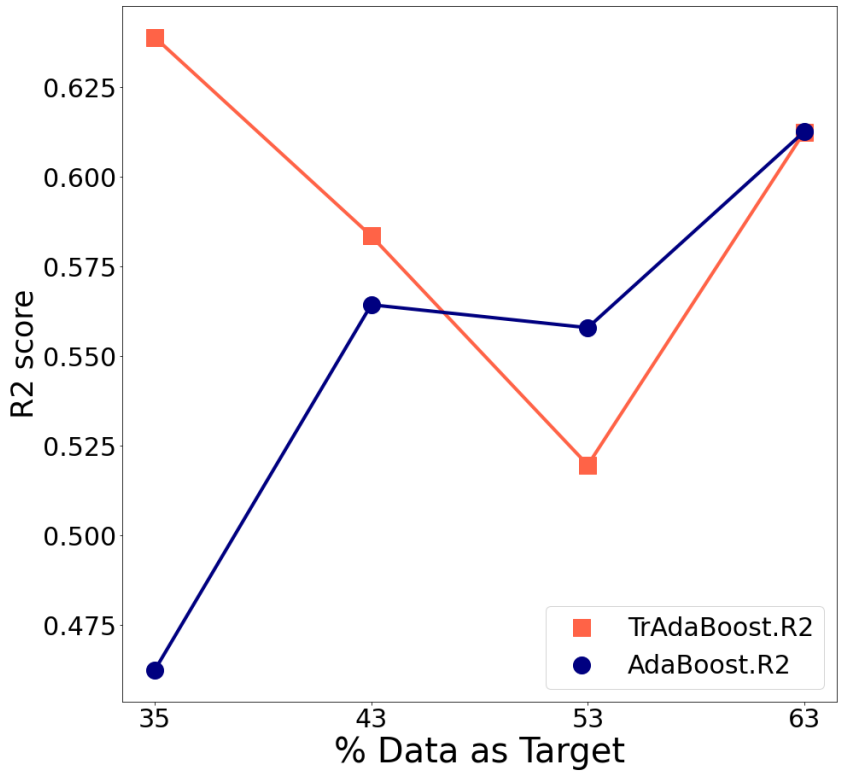

Figure 3.1: Negative transfer in TTR2 is induced as a result of increasing the target sample size from 35% to 63% of the total training data. The baseline algorithm is AdaBoost.R2. For a larger target sample size, the baseline performs better than TTR2

Boosting for Regression

When boosting is used for regression, it faces the problem of reweighing incorrectly predicted instances. AdaBoost.R2 [50] introduces the concept of adjusted error to reduce the effect of an arbitrarily large number for the predicted error; defined as,

$$e'_i = \frac{e_i}{max_{i=1}^{n}|e_i|} \tag{3.1}$$

$$\text{where,} \quad e_i = |y(x_i) - h(x_i)| \tag{3.2}$$

where e_i denotes the predicted error on the hypothesis h_t and i are the number

of training instances. The inflation in the error is normalized by maximizing the error over the instances in the previous iteration. The weights of the instances in AdaBoost.R2 are updated as,

$$w_i^{t+1} = \frac{w_i^t \beta_t^{1-e'_{it}}}{Z_t} \tag{3.3}$$

$$\text{where,} \quad \beta_t = \eta_t / 1 - \eta_t \quad \text{and} \quad \eta_t = \sum_{k=1}^{n} w_i^t e_i^t \tag{3.4}$$

where Z_t is normalizing constant, t is current iteration.

Boosting for Transfer Learning

TrAdaBoost [40] is a classification boosting framework that applies transfer learning to compensate for a lack of training instances for the target dataset. The source and target data instances are merged to form the training data for the TrAdaBoost, and in each iteration, the weights of the instances are adjusted such that the misclassified target instances have their weights increased, whereas the misclassified source instances have their weights reduced, in order to reduce their impact towards the model learning. However, this may lead to model over-fitting, and reduction in the variance of the training model, therefore negatively affecting the model generalizability [212].

Boosting for Regression Transfer

TrAdaBoost.R2 [169] (as shown in algorithm 1) builds upon TrAdaBoost [50] for regression problems, using adjusted error over residuals and reweighing of the instances. The improved version, called two-stage TrAdaBoost.R2 (TTR2), is divided into two stages. The first stage involves gradually reducing the weights of the source instances until a certain cross-validation threshold is achieved. In the second stage, weights of the source instances are frozen while the weights of the target instances are

updated as in AdaBoost.R2. The bi-update methodology for TTR2 helps reduce the skewness produced due to source instances. This mostly happens in the cases when source sample size is very large compared to the target sample size, which consequently makes the model learning biased towards the source domain.

3.2.2 Variants of Regression Transfer

Algorithm 1: Two-Stage TrAdaBoost.R2 (Pardoe and Stone)

Input: Training set T with source instances $1 \dots n$ and target instances $1 \dots m$, number of estimators N, base learner *learner*, iterations S

Output: Final hypothesis h_f

for $t \leftarrow 1$ *to* S **do**

Call AdaBoost.R2' with T, w^t, N, *learner* to obtain $model_t$, where AdaBoost.R2' is similar to AdaBoost.R2 except weights of source instances $1 \dots n$ in T are never modified.

Get a hypothesis h_t for T and distribution w^t, and calculate the adjusted error e_i^t for each instance as in AdaBoost.R2.

Update the weights as:

$$w_i^{t+1} = \begin{cases} \frac{w_i^t \beta_t^{e_i^t}}{Z_t}, & \text{if } 1 \leq i \leq n \\ \frac{w_i^t}{Z_t}, & \text{if } n < i \leq (n+m) \end{cases}$$

where Z_t is the normalizing constant and β_t is chosen such that the resulting weight of target instances is

$$\frac{m}{(n+m)} + \frac{t}{(S-1)} \left(1 - \frac{m}{(n+m)}\right).$$

end

return h_f where $f = \arg\min_i \text{ error}_i$

Pardoe et al. [169] introduced two categories of transfer learning algorithms. The first category contains algorithms that choose the best hypothesis from a set of experts, each representing the models for the corresponding source dataset. This category includes algorithms such as *ExpBoost.R2* and *Transfer Stacking*. Algorithms in the second category, which include TrAdaBoost.R2 and TTR2, use the grouped

source and target datasets to perform boosting. Since boosting methodologies involve instance reweighing, they fall under the category of transfer learning algorithms that use domain adaptation. This is especially useful and applicable for real-world datasets with dissimilar domain distributions. Hence, such domain adaptation transfer methodologies help in reducing the burden of maintaining expert systems [183]. Apart from the boosting methodologies, the varying domain adaptation approaches include using a kernel-employing Gaussian process [26] for source instance modification or kernel ridge regression, and discrepancy minimization for domain adaptation [38]. Similar to importance sampling [165], several studies [157, 71] have used importance weighting of source instances to improve inference for transferring knowledge. Transfer methodologies using approaches similar to active learning, such as [44] (employing modeling structure with second-order Markov chains), as well as a variety of deep learning approaches [13, 43], are indicative of the usefulness of active learning in the form of importance sampling as a viable technique to be picked up by ITL methodologies.

The ITL models – KMM.TL*(Kernel Mean Matching) [100],* KLIEP.TL*(Kullback–Leibler Importance Estimation Procedure) [195] used in this chapter have been also employed in RQ2. We elaborate on theri mathematical formulation in detail in that chapter.*

Importance Weighted Kernel Ridge Regression

The IW-KRR.TL (Importance Weighted Kernel Ridge Regression) [71] method expands upon the Importance Weighted Least Squares (IWLS) whereby it incorporates kernel ridge regression (KRR), which allows it to capture existing non-linear relationships/dependencies in the dataset [71]. In IW-KRR.TL, each source sample is re-weighted to to resemble the target samples, for an improved prediction accuracy. It solves the following ridge regression objective:

$$\min_{w} \sum_{i=1}^{m} \hat{p}(x_i^T)/\hat{q}(x_i^S) \cdot \left(y_i^S - f(x_i^S)\right)^2 + \lambda \|f\|_K^2,$$

where x_i^S and y_i^S represent source domain features and target respectively, and $\hat{p}(x_i^T)/\hat{q}(x_i^S)$ represents the importance weight for each source sample, also the density estimation ratio calculated using the target $\hat{p}(x_i^T)$ and source $\hat{q}(x_i^S)$ distributions. $\lambda\|f\|_K^2$ is a regularization term, where $\|f\|_K$ is the norm in the Reproducing Kernel Hilbert Space (RKHS) defined by a kernel function K, which allows to manage the model complexity. The key components of the methodology are:

- **Dataset Shift Handling:** Where the IW-KRR.TL approach directly addresses dataset shift between the source and target distributions by reweighing source samples to aligning them with the target sample.
- **Density Estimation:** The importance weights are calculated using density ratio estimation, $\frac{\hat{p}(x^T)}{\hat{q}(x^S)}$, with techniques such as Kernel Density Estimation (KDE) or alternative ratio approximation methods.
- **Kernel Methods:** The kernel ridge regression framework allows handling complex, non-linear relationships/dependencies present in the data such that it defines a function, f in a high-dimensional feature space induced by the kernel function, $K(x, x')$.

3.2.3 Importance Sampling

Importance sampling is a methodology based on the concept that certain instances of the source dataset are more similarly distributed to the instances in the target dataset and thus should be sampled for learning optimal transfer models [110, 165, 232]. Zhao et al. [250] introduce stochastic optimization for importance sampling of non-transfer learning problems, to reduce variance and improve convergence. Elvira et al. [55, 56]

utilize gradient-based learning whereas Bullago et al. [23] and Schuster et al. [189] apply Monte Carlo methods to apply adaptive importance sampling. Salaken et al. [185] present a seeded sampling technique for transfer learning that we extend to form the variance sampling component used by our algorithm, STrAdaBoost.R2. Their work introduces an algorithm to cluster the source domain instances which are then translated to limited target domain instances for knowledge/domain adaptation. In the following section, we describe how we utilize the concept used by seeded sampling for cherry-picking instances from the source domain for the purpose of introducing variance in the target dataset.

3.2.4 Complexity of Distribution

Table 3.1: Dataset Statistics [Tr: Training, Tt: Test, P_C^M: predictor] and Complexity

	Concrete	Housing	Auto	Ailerons	Elevators	Abalone	Kinematics	C.Activity
Shape	(1030, 9)	(506, 14)	(392, 8)	Tr: (7154, 41) Tt: (6596, 41)	Tr: (8572, 19) Tt: (7847, 19)	(4177, 9)	(8192, 9)	(8192, 22)
Target	Strength	medv	mpg	goal	Goal	Rings	y	usr
P_C^M	Cement	nox	h.power	*None*	*None*	weight	theta7	pgin
C_{FE}	0.66	0.39	0.51	0.47	0.59	0.69	**0.70**	0.36
D_L	0.20	0.29	0.24	0.26	0.32	0.27	0.19	**0.36**
D_I	0.71	0.90	0.58	0.68	0.59	0.51	**1.08**	0.58

Domain-agnostic characterizations of dataset complexity are surprisingly uncommon. Fernandez et al. [63] present a characterization based on Shannon entropy, but this does not extend to the continuous, often real-valued domains of many real-world datasets [20]. Other intuitive measures such as sorting datasets by the number of features or self-similarity do not reliably capture types of datasets that we observed as being especially prone to negative transfer. The heterogeneity and complexity of datasets usually determine the model performance. While the heterogeneity of real-world datasets can be outlined as a factor of their multi-source and spatiotemporal character, this might not be true for their complexity. Ho et al. [92] proposed metrics to measure complexity for classification datasets. Maciel et al. [142] extended that

work for regression datasets which stemmed from the work done by Lorena et al. [137] that utilizes meta-features as a measure of complexity. In the following sections, we discuss and apply the measures provided by Maciel et al. [142] to characterize the complexity of regression datasets.

Collective Feature Efficiency (C_{FE}): Correlation Measure

The correlation measure determines the highly correlated predictor to the target variable and fits a linear regressor to find its residuals. All the instances having residual less than a certain threshold ($\epsilon \leq 0.1$) are discarded and the remaining instances are used to determine the next highly correlated predictor. The process is repeated until the complete feature space has been visited. Maciel et al. [142] describes the measure as the Collective Feature Efficiency (C_{FE}) which is expressed as,

$$C_{FE} = 1 - \sum_{k} \frac{N_k}{N}$$

where N_k is the number of instances that are removed (using the set threshold), N is the total number of instances and k is the feature. Higher values for C_{FE} indicate more complex problems.

Distance from Linear Function (D_L): Linearity Measure

The linearity measure sums the absolute values of residuals when a multiple linear regressor is used as the learner [142]. It is expressed as a distance measure (D_L) and is quantified as,

$$D_L = 1 - \sum_{i=1}^{N} \frac{R_i}{N}$$

where R_i are the residues and N is the sample size. Lower values indicate a simpler distribution.

Input Distribution (D_I): Smoothness Measure

The smoothness measure determines the smoothness of the distribution by ordering the predictor values in ascending order with regard to the output variable. It then finds the distance (L2 Norm) between each pair of instances [142]. Lower values mean a simpler distribution, indicating that the instances in input space are closer to each other, leading to a smooth distribution. It is expressed as,

$$D_I = \frac{1}{N}\sum_{i=2}^{N}\|\mathbf{x_i} - \mathbf{x_{i-1}}\|$$

where N is the sample size and $\|.\|$ is the Euclidean distance.

We break down the transfer learning process by understanding the mathematics behind the transfer of knowledge in the context of machine learning. We then look into instance transfer learning methodologies which are utilized when the source and the target domains have the same feature space. During our preliminary experiments, we noticed how transfer learning methodologies are over-fitted for each dataset, and their performance varied with the complexity of the dataset distribution. We improved an existing ITL methodology – TrAdaBoost.R2 to make it more generalizable as well as define metrics to measure the complexity of dataset distribution.

3.3 Methodology

3.3.1 Problem Definition:

Given source and target datasets, such that their instances are denoted by x^T and x^S respectively. Hence, the target dataset is denoted as $X^T = \{x_1^T, x_2^T, ..., x_m^T\}$ for m instances and source dataset is denoted as $X^S = \{x_1^S, x_2^S, ..., x_n^S\}$ for n instances. Similarly, the target output dataset is denoted as $Y^T = \{y_1^T, y_2^T, ..., y_m^T\}$ and the source output dataset is denoted as $Y^S = \{y_1^S, y_2^S, ..., y_n^S\}$. The target domain suffers from

significant data deficiency and dissimilarity of distribution compared to the source domain. Our goal is to find a transfer learning approach that can use the source domain instances as leverage for building the prediction model as well as avoiding negative transfer. The transfer learning algorithm should perform consistently well on varying domain distributions with differing complexities.

3.3.2 Approach:

Algorithm 2: S-TrAdaBoost.R2

Input: Labeled data sets X^S (size n) and X^T (size m), number of estimators N, cross-validation folds F, iterations S, base learner *learner*, learning rate α

Output: Final hypothesis h_f

Importance Sampling: Get updated source dataset X^{ES} with p instances from X^S most similar to X^T.

Variance Sampling: Get updated target dataset X^{VT} with q instances using k-Center Sampling on X^T.

Initialize: Set initial weight $w^1 = 1/(p+q)$.

for $t \leftarrow 1$ *to* S **do**

  Call AdaBoost.R2 with N estimators and *learner* to get hypothesis h_t.

  foreach i *in* $1, \ldots, p+q$ **do**

    $e_i = |y(x_i) - h(x_i)|/J$, where $J = \max_{i=1}^{p+q} |e_i|$

  end

  Set $\bar{\beta}_t = \eta_t/(1-\eta_t)$, where $\eta_t = \sum_{i=1}^{p+q} w_i^t e_i^t$ and $\beta_t = \frac{q}{p+q} + \frac{t}{S-1}\left(1 - \frac{q}{p+q}\right)$

  foreach i *in* $1, \ldots, p+q$ **do**

$$w_i^{t+1} = \begin{cases} \frac{w_i^t \bar{\beta}_t^{e_i^t} \alpha}{Z_t} & \text{if } 1 \le i \le p \\ \frac{w_i^t \beta_t^{1-e_i^t} \alpha}{Z_t} & \text{if } p < i \le p+q \end{cases}$$

  end

  where Z_t is the sum of sample weights.

end

return h_f where $f = \arg\min_i \text{ error}_i$

STrAdaBoost.R2 is a transfer regression boosting algorithm which builds a model, $h_f : X \to Y$, such that h_f is the final learned hypothesis of the ensemble of hypotheses over the learning iterations, using the training data which is a combination of source

and target datasets that share a similar feature space but have dissimilar distributions. Hence by this definition, the combined training dataset (source + target) can be denoted as $\{(x, y) \| x \in X^T \cup X^S, y \in Y^T \cup Y^S$ and $X^T, X^S, Y^T, Y^S \in R^d\}$ where d represents the feature space of the source and target domain.

3.3.3 STrAdaBoost.R2

To improve the performance of TTR2, we present STrAdaBoost.R2 as shown in algorithm 2. There are two main areas where STrAdaBoost.R2 diverges from its predecessor, TTR2; the first is applying importance sampling, and the second is the weight update strategy for STrAdaBoost.R2, which differs from the TTR2. In the following subsections, we elaborate upon these differences as well as determine the time complexity of STrAdaBoost.R2.

Sampling

In order to improve the prediction accuracy, S-TrAdaBoost.R2 initially samples the source dataset, X^S, to obtain optimal representative instances, i.e. similar instances to the target dataset, X^T. Hence, before merging the source-domain and target-domain samples, we apply importance sampling to carefully select favorable source-domain instances. We utilize a greedy approach for calculating the distance between the source and the target instances. Such an importance sampling can be achieved by utilizing distance measures (Euclidean, Manhattan, and more) as well as alternative methodologies utilizing gradient-based and similarity-based sample selection [55, 56, 23, 189]. For our experiments, we use the Euclidean distance (L2 norm). Hence, we find the set $X^{ES} \subset X^S$ such that,

$$X^{ES} = \|\mathbf{x_i^S} - \bar{\mathbf{x}}^\mathbf{T}\| \qquad \forall x_i \in X^S$$

where $\bar{x}^T$ is the mean of target instances, $\|.\|$ is the Euclidean distance, and $|X^{ES}| = |X^S|$, i.e. they share the same cardinality. We select the top p instances from X^{ES} for the source dataset, which reduces the source dataset size to $X^K = \{x_1^K, x_2^K, ..., x_p^K\}$ such that $p << n$ and discard the remaining $(n - p)$ instances since they failed the similarity testing threshold.

Furthermore, to improve the generalizability of the prediction model, we also induce variance in the target dataset whereby source instances most similar to the target instances are added using the k-center sampling, an approach presented in algorithm 3. Including the most similarly distributed source samples in the target dataset improves the fit for the regressor since S-TrAdaBoost.R2 focuses more on target instances than the source instances. These similarly distributed source samples act as noise for the target distribution and thereby improve the generalization error. Even though TTR2 tries to mitigate this using its two-stage source instance penalizing process, we found that reducing the source sample size using importance sampling, as well as performing variance sampling, allows S-TrAdaBoost.R2 to perform better compared to its predecessor.

Algorithm 3: k-Center Sampling

Input: X^T, Y^T, X^S, Y^S
Output: Labeled dataset X^{VT} (size k).
1 Find $X^C \subset X^S$ such that $X^C = \{x_1^C, x_2^C, ..., x_k^C\}$ has k samples, obtained using k-means clustering on X^S.
2 Initialize $X^E = \phi$ (Empty-set)
3 **for** $x^C \in X^C$ **do**
4 | Find x^T such that $\forall x^T \in X^T \;\; \min(\|X^C - x^T\|)$
5 | $X^E \cup \{x^T\}$
6 **end**
7 Repeat steps 3 to 5 and obtain set $X^{VT} \subset X^S$ closest to instances in set X^E.
8 **return** X^{VT}

k-center sampling k-center sampling is an unsupervised approach that returns k centroids, where k is equal to the number of source instances in the set, X^S

(algorithm 3). We employ k-center sampling in our methodology to introduce noise in the target dataset, in order to increase its variability. After the selection of centroids, the target instances closest to these centroids are selected as the representative target set, X^C. The source instances most similar to the representative target set are chosen as the final subset, X^{VT}, for inclusion into the target dataset. The k-center sampling methodology is presented in algorithm 3. The final size of the target dataset is, $q = n + k$. For the k-center sampling, the time complexity is $O(N^2)$ as a result of using the k-means clustering for calculating the closeness. Hence, the sampling pipeline produces a new source dataset (due to Importance Sampling) and a new target dataset (due to Variance Sampling) as X^{ES} and X^{VT} respectively.

Weight Update Strategy

We present S-TrAdaBoost.R2 in algorithm 2, where we hypothesize that by updating the target weights more aggressively, the prediction model is able to mitigate the source distribution bias. This is especially useful for dissimilar source and target domain distributions, as well as when $|X^S| >> |X^T|$. We also note that S-TrAdaBoost.R2 does not employ AdaBoost.R2' [169], a modified version of AdaBoost.R2 where the weights of source instances are frozen and the weights of target instances are updated based on the reweighing approach used by AdaBoost.R2. However, applying highly focused domain adaptation by freezing weights of source instances can greatly reduce the generalizability of the model, as performed in the previous technique, TTR2. For this reason, our approach penalizes both the source domain and target domain instances allowing for a balanced weighing. Hence, in S-TrAdaBoost.R2, the hypothesis is obtained by using the AdaBoost.R2 methodology initially. The weights for the instances are then updated iteratively using the following weight equation,

$$w_i^{t+1} = \begin{cases} \frac{w_i^t \bar{\beta}_t^{e_i^t} \alpha}{Z_t}, & 1 \leq i \leq p \\ \frac{w_i^t \beta_t^{1-e_i^t} \alpha}{Z_t}, & p \leq i \leq (p+q) \end{cases}$$

In the above equation, $\bar{\beta}_t = \eta_t / 1 - \eta_t$ such that $\eta_t = \sum_{k=1}^{(p+q)} w_i^t e_i^t$, and $Z_t = \sum_{k=1}^{(p+q)} w_i \beta_t$ indicates the sum of sample weights. For the above weighing strategy, the source domain instances are penalized more aggressively with both β and e_i depending on instance residual compared to the target domain instances with constant β. This allows for a balanced weighing where both domain instances are penalized with the target instance weighing being slower compared to the source instance weighing to balance the skewness caused by a large number of source instances. Hence, although the source instances are penalized more than target instances, the instance weighing is still not as aggressive as in the predecessor methodology, TTR2 which can lead to overfitting on the dataset.

Time Complexity for S-TrAdaBoost.R2

The time complexity of the S-TrAdaBoost.R2 can be divided into *four* parts:

1. Time complexity of importance sampling (O_1)
2. Time complexity of the weak hypothesis (O_2)
3. Time complexity of computing the error rate in S-TrAdaBoost.R2 (O_3)
4. Time complexity of the second stage of S-TrAdaBoost.R2 (O_4)

For S iterations, time complexity can be defined as $O(S * (O_2 + O_3 + O_4))$. For our experiments, we chose a decision tree as the base learner. The time complexity for creating a decision tree is $O(d * N^2 * logN)$ (O_2), where d is the dimension of the dataset, N is the number of samples, and each decision is taken in $O(logN)$ time. The time complexity of computing adjusted error combined with the weight update

process (O_3), does not increase more than $O(N^2)$. Finally, the time complexity of computing the second stage of the S-TrAdaBoost.R2 is similar to producing a weak hypothesis (O_4). Hence, the time complexity over S iterations is,

$$\begin{aligned} O(S * (d * N^2 * logN + N + d * N^2 * logN)) = \\ O(2 * S * d * N^2 * logN + S * N)) \\ = O(S * d * N^2 * logN)) \end{aligned}$$

For the k-center sampling, the time complexity is $O(N^2)$ for calculating closeness using the k-means clustering, as well as using Manhattan distance for finding the most similar source instances. Hence, the total time complexity for S-TrAdaBoost.R2 can be calculated as,

$$O(S * d * N^2 * logN + N^2) = O(S * d * N^2 * logN)$$

3.4 Evaluation

For our experiments, we evaluate S-TrAdaBoost.R2 against other competitive transfer learning methodologies such as TTR2 (Two-stage TrAdaBoost.R2) [169], KMM.TL(Kernel Mean Matching) [100], KLIEP.TL(Kullback–Leibler Importance Estimation Procedure) [195] and IW-KRR.TL (Importance Weighted Kernel Ridge Regression) [71] known to perform well for regression-based instance transfer learning problems. Since TTR2 is the predecessor for S-TrAdaBoost.R2, we define it as the baseline algorithm for comparison. The decision tree regressor was chosen as the base learner for these methodologies. For TTR2 and S-TrAdaBoost.R2, the following values were considered: S (no. of steps) = 30, F (CV-folds) = 10, learning rate = 0.1 and a *squared loss*. Similar values were used by Pardoe et al. [169] for their study on regression boosting. For the remaining algorithms, we used the default values for the

parameters. The values were chosen to maintain generalizability of the predictions across the algorithm. They were derived using multiple experiments and iterations involving parameter tuning, and were judged to not be biased towards a single model to the best of our knowledge. The results along with the ablation study are presented in the following sections.

3.4.1 Datasets

We chose 8 standard regression datasets from the UCI machine learning repository [6] as shown in Table 3.1. UCI datasets were divided into *source*, *target*, and *test* sets using the splitting methodology used by Pardoe et al. [169]. The splits were made by identifying the feature moderately correlated with the target variable, which allowed for concepts to be significantly different from each other. The *first* split was considered as the target dataset and the remaining splits as the source dataset. This was done so that the source sample size would be higher than the target sample size. The target dataset was further split into training and testing datasets using a k-fold split over 20 iterations. Our initial study showed that the root mean squared loss (RMSE) on concrete, housing, and automobile datasets were moderately varied for such a division which allowed for robust predictions since it incorporated both generalizability for the models, as well as lesser noise. Hence, we further extended the splitting methodology to other datasets – abalone, kinematics, and computer activity. For ailerons and elevators datasets, the UCI repository already consisted of a testing dataset. We took very few target instances so that the remaining larger dataset could be used as the source dataset, which in turn imitates a real-world transfer learning problem. Table 3.1 shows the dataset statistics including their size, target variable, and predictor used for correlation splitting. Although Concrete, Housing, and Automobile are small sample datasets, they were used to imitate the study by Pardoe et al. [169]. We compensated for this imbalance using other large sample datasets with varying heterogeneity. The

complexity evaluation in Table 3.1 shows the complexity of dataset distributions based on variance (C_{FE}), smoothness (D_I), and linearity (D_L). For each measure, a higher value indicates a more complex distribution. We observe that *Kinematics* has the highest complexity (2 out of 3 times) when compared to the other datasets.

3.4.2 Ablation Study

We perform an ablation study where the importance sampling technique is applied individually to each transfer learning methodology. The goal of this study is to induce fairness in comparison, given the modular nature of importance sampling. Sampling is a two-phase methodology that includes variance sampling and importance sampling. The variance Sampling includes sprinkling the target dataset with source instances in order to introduce noise and increase the variance of the distribution. For the concrete, housing, and automobile datasets, variance sampling was not applied due to the low sample size. The importance sampling on the other hand uses similarity measuring to find the source instances most similar (important) to the target instances. The ablation study exploits importance sampling for all the methodologies and variance sampling for larger datasets.

3.4.3 Results

We implemented the experiments on an HPC cluster with 16 processors and 128 GB RAM. Any required short supplemental processing was performed on personal laptops with half the number of processors and RAM. The number of cross-validation folds was 20 for the datasets. The distribution of prediction values is shown in the box-plot Figure 3.2. We observe that S-TrAdaBoost.R2consistently performs well, with low RMSE as well as a high R-squared score. However, this is not true for other methodologies, especially IW-KRR.TL and TTR2 which, although they sometimes outperform S-TrAdaBoost.R2, also fluctuate highly in their performance. Example

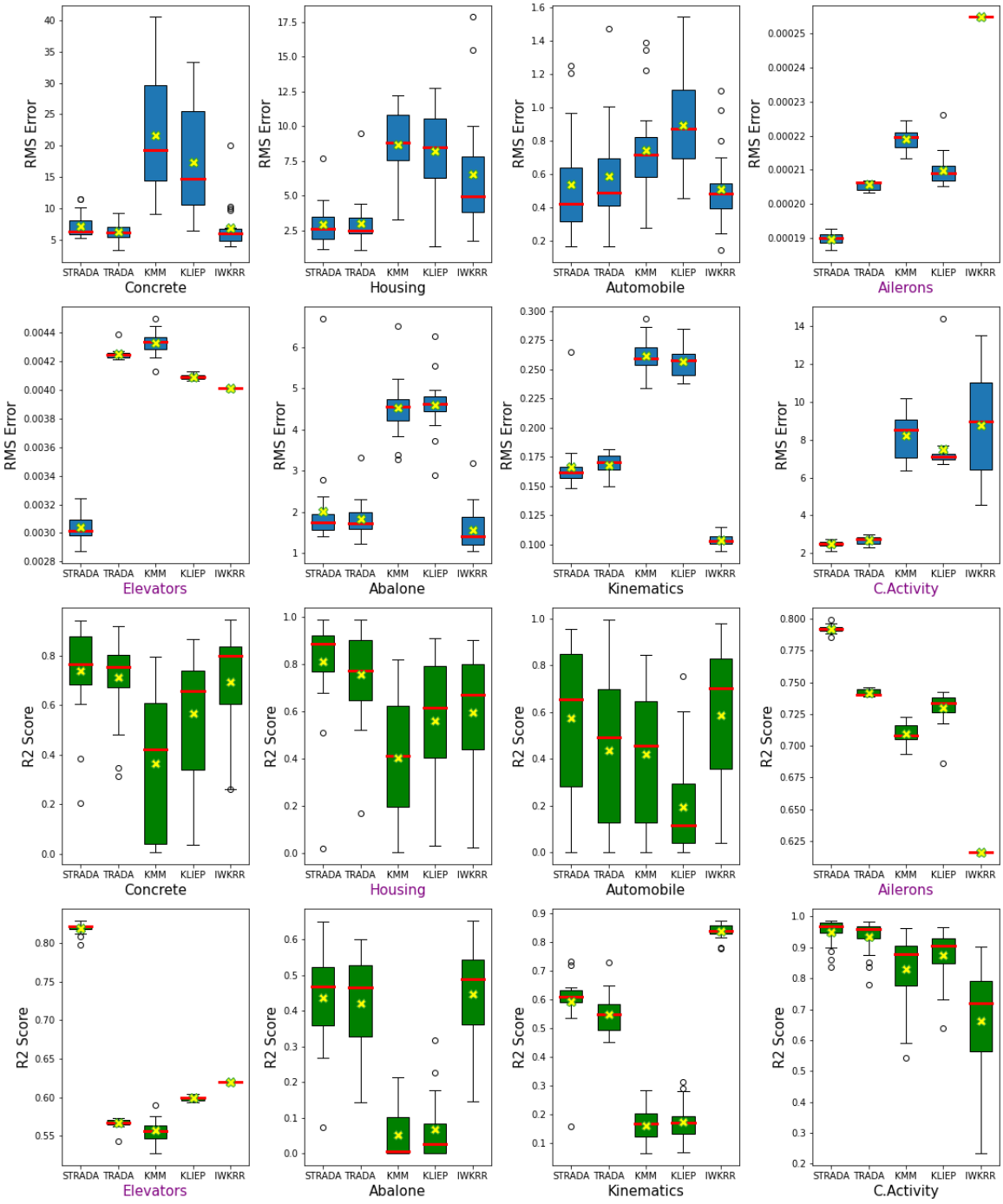

Figure 3.2: Comparison of transfer learning algorithms– TRADA: TTR2, STRADA: S-TrAdaBoost.R2, KMM: KMM.TL, and KLIEP: KLIEP.TL, IWKRR: IW-KRR.TL, where the RMS error and R-squared score is calculated over 20 iterations. The Interquartile Range (IQR), mean value (marker: yellow "X"), and median value (marker: red line) for each algorithm over the iterations have been highlighted. The datasets for which S-TrAdaBoost.R2 performs particularly well are marked as well (marker: purple).

IW-KRR.TL is the most optimal model for automobile, abalone, and kinematics datasets as observed through its mean RMSE and R-squared values. But it is not consistent in its performance as observed for computer activity, ailerons, and elevators datasets, where it fluctuates highly in its mean and variance over the iterations. However, S-TrAdaBoost.R2 performs consistently well for all of the datasets and comes a close second in the kinematics dataset, where IW-KRR.TL outperforms the competing methodologies by a high margin. Similarly, for TTR2, we observe that it performs well (RMSE score) on concrete and abalone datasets compared to S-TrAdaBoost.R2, but its performance is not consistent as observed for ailerons and elevators datasets. We consider TTR2 to be our baseline algorithm for this study primarily because it is the predecessor of S-TrAdaBoost.R2, and observe that S-TrAdaBoost.R2 outperforms TTR2 75% of the times in the case of loss measure, and 100% when measured for correlation accuracy.

Table 3.2: Ablation Study

	Ailerons		Elevators		Abalone		Kinematics		C.Activity	
	RMSE	R^2	RMS	R^2	RMS	R^2	RMS	R^2	RMS	R^2
TRADA	0.00023	0.65	0.0042	0.38	2.14	0.40	0.18	0.47	2.98	0.92
STRADA	**0.00018**	**0.79**	0.0030	**0.81**	2.02	**0.43**	0.18	0.51	**2.48**	**0.94**
KMM	0.00029	0.46	0.0049	0.31	2.73	0.06	0.27	0.08	11.30	0.17
KLIEP	0.00026	0.58	0.0043	0.42	2.76	0.10	0.26	0.10	11.09	0.22
IWKRR	0.00025	0.63	**0.0021**	**0.81**	**1.99**	0.41	**0.10**	**0.84**	8.77	0.66

Considering that the importance sampling is a pre-domain adaptation methodology and should not be limited to just S-TrAdaBoost.R2, we conduct an Ablation study as shown in Table 3.2. We observe minimal improvement in the performance of TTR2 and IW-KRR.TL and find that S-TrAdaBoost.R2 performs consistently well (4 out of 5 times). Table 3.2 shows that IW-KRR.TL has competitive scores with regard to S-TrAdaBoost.R2, however, it has the same inconsistent performance as observed in the comparative study presented in Figure 3.2. Also, TTR2 does not show any improvement except for a similar RMSE score to S-TrAdaBoost.R2 for the kinematics dataset. However, IW-KRR.TL easily outperforms all other methodologies

for the kinematics dataset. It should also be noted that in both studies, the remaining algorithms KMM.TL and KLIEP.TL performed quite poorly compared to the other methodologies and showed no apparent sign of improvement in either case. Hence, we can say that S-TrAdaBoost.R2 has shown itself to be consistent among all the measures, adapting more robustly to more complex and varying distribution datasets.

3.5 Discussion

Since S-TrAdaBoost.R2 is a successor to TTR2, we use TTR2 as the baseline methodology and observe that S-TrAdaBoost.R2 outperforms it 7 out of 8 times during the comparative study. We also note that TTR2 shows no significant improvement during the ablation study. This justifies the steady performance of S-TrAdaBoost.R2, where it consistently has optimal RMSE and R-squared scores during the comparative and ablation studies. The ablation study is used to justify how importance sampling is useful when combined with the learning methodology for S-TrAdaBoost.R2. This is due to the balanced weighing complimenting the source domain sampling methodology. We find that for relatively complex datasets such as concrete, elevators, kinematics, and c.activity, S-TrAdaBoost.R2 performs well on most of them (3 out of 4 times), falling short only in the case of the kinematics dataset when compared to IW-KRR.TL methodology.

It should be noted that both the training error and the generalization error of a similar problem space have been analyzed thoroughly in Schapire et al. [188], and this analysis is further known to apply to TrAdaBoost.R2 [169], a predecessor to S-TrAdaBoost.R2. The objective function for transfer learning involves minimizing the loss, $min_h\{\mathcal{L}(h) + \lambda\eta\}$, where η is the regularization function, and λ is the regularization constant for the loss function $\mathcal{L}$. We hypothesize a function $h \in H$ that maps training instances, predictor $x \in X$ to target $y \in Y$ in the target domain T_T.

Hence, the instance transfer methodology tries to minimize the weighted loss of target and source domain [212] ($\mathcal{L}(h) = \mathcal{L}_T(h) + \mathcal{L}_S(h)$). Since S-TrAdaBoost.R2 relies on using AdaBoost.R2 unlike TTR2 [169], it has increased generalizability as it avoids overfitting while assigning balanced source and target weights.

While S-TrAdaBoost.R2 has improved generalizability by utilizing balanced reweighing and sampling methodologies, it can, however, be limited by the computational overhead and poorly strategized implementation of the sampling methodologies. The importance sampling methodology can reduce the performance of transfer learning if the threshold for sampling is high i.e. very few source-domain instances are selected. Furthermore, for large source-domain datasets ($> 10^5$), sampling methodologies (importance sampling and variance sampling) cause additional computational overhead. Hence, while these methodologies are simpler to implement, the initial and sampled instances affect the performance of our approach.

3.6 Summary

In this objective, we introduced S-TrAdaBoost.R2, which uses importance sampling combined with an unrestricted weight update strategy to improve performance for instance transfer learning by an average of 12% across all datasets, and 13% in sufficiently complex datasets when compared to its predecessor, TTR2. To better characterize the datasets that S-TrAdaBoost.R2 performs well on, we utilize complexity measures [142], C_{FE}, D_L and D_I that employ feature correlation and fitting a linear regressor to compute the complexity for the datasets. Hence, we can conclude that S-TrAdaBoost.R2 would be well suited for complex real-world datasets that vary in distribution, as well as the uniformity of features. Hence, the functional improvements we propose to TTR2 are modest enough that we expect S-TrAdaBoost.R2 as a replacement for TTR2 and other instance transfer methodologies in scientific data

analysis pipelines.

In the following objectives, we focus on spatiotemporal datasets, which include consequential domains such as pollution prediction, weather, and forest fires forecasting, and more. These domains are dependent on ground stations for successfully collecting and forecasting data. Therefore, the transfer learning problem for these domains suffers from multiple challenges such as *transfer across space*, *transfer over time*, and *spatiotemporal transfer learning* which we explain in the following sections. As mentioned previously, the pollution prediction domain is particularly significant for us and a constant motivation for this dissertation.

Chapter 4

Transfer across regions w/ similar feature space

This objective focuses on transfer learning for spatiotemporal data for the pollution prediction use-case. Our problem focus is estimating $PM_{2.5}$ for countries/regions with less ground-sensors. We employ instance transfer learning models for knowledge transfer from countries/regions with large number of ground-sensors. Additionally, the source and target datasets belong to two spatially seperated regions (cities/states/countries) with varying distributions as well as meterological and topographical diversity. Since, the source and target domains share the same feature space, we don't employ feature standardization in this objective. The two regions are also geographically within the same hemisphere (i.e. within a country), reducing the complexity of transfer.

The following objective **(RQ3)** picks up on this objective and applies transfer learning for datasets with dissimilar feature space and geographically distant regions.

The two objectives, **(RQ2)** and **(RQ3)** were combined and published as a single paper in the conference – European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2024) [81].

4.1 Rationale

Air pollution, especially atmospheric aerosols smaller than 2.5 micrometers *i.e.* $PM_{2.5}$ poses a significant concern to public health [191]. Emissions from vehicles [113], wildfires [46], and industrial processes [54] are major contributors to high $PM_{2.5}$ levels. Current approaches for measuring $PM_{2.5}$ involves using either remote sensing methodologies [14] or ground sensors [7]. While satellite-based remote sensing methodologies are a low-cost way to measure $PM_{2.5}$, however, their data collection is affected by factors like cloudy weather and high surface reflectance, thereby significantly reducing the accuracy of measured $PM_{2.5}$ levels [14]. Alternatively, installing $PM_{2.5}$ ground sensors yields highly accurate data as these sensors employ gravimetric data collection methodologies [7]. However, due to their high installation and maintenance costs [118], it is challenging to scale them in developing countries [47], creating an imbalance of *data-rich* (developed) and *data-poor* (developing) regions with $PM_{2.5}$ data for air pollution estimation.

Transfer learning (TL) can ameliorate this situation by utilizing *data-rich* (source data) regions to learn a prediction model on *data-poor* (target data) regions [165]. Prior research on estimating $PM_{2.5}$ through TL is geared towards time-series forecasting where the model learns historical data of an observed location (sensors) and forecasts the horizon (*i.e.* future values) for the observed locations [67, 140, 233, 234]. Therefore, these models cannot estimate the $PM_{2.5}$ levels for locations where historical data is unavailable [205]. Alternatively, one can employ *Instance transfer learning* (ITL) models that avoid the limitations of time-series forecasting models by not relying on continuous temporal data [71, 79]. ITL models reweigh source domain samples based on the target domain and subsequently combine the two domains.

Unfortunately, ITL models are limited in estimating $PM_{2.5}$ as they overlook the *spatial and semantic correlations* in the datasets. $PM_{2.5}$ estimation data is uniquely heterogeneous and complex, containing topographical, meteorological, and

geographical features. These features exhibit *spatial* autocorrelations (dependencies), *i.e.* nearby locations tend to have similar $PM_{2.5}$ levels, as well as *semantic* correlations (dependencies), *e.g.* locations with similar meteorological and topographical conditions exhibit similar $PM_{2.5}$ levels with high likelihood [125]. Spatial dependencies are prevalent within a domain, whereas semantic dependencies will likely arise when combining two domains (case for ITL). We call this transfer problem as *spatial* transfer learning.

In this objective, we solve *spatial* transfer learning to improve $PM_{2.5}$ estimation by allowing source and target data points to learn from each other in the combined domain space. We achieve this by introducing a new feature called *Latent Dependency Factor* (LDF) in both the source and target datasets to bridge the gap between the two domains. To generate LDF, we first learn a cluster of similar (spatially and semantically similar) data points for each sample, which are fed to our novel two-stage autoencoder model. The first stage, *encoder-decoder*, aims to learn a latent representation from the combined feature space of the cluster, while the second stage, *encoder-estimator*, learns from the target label ($PM_{2.5}$ value). The LDF is highly correlated to the target (dependent) variable and contains learned dependencies from both domains. To illustrate the benefits of LDF, we utilize real-world $PM_{2.5}$ data for the United States and Lima city in Peru. Our experiments include a comparative analysis of ML and TL models within the US boundaries, where we observe a 19.34% improvement in prediction accuracy over baseline models. We also present a qualitative analysis showcasing how our model captures larger estimation patterns better than the competetive baselines. In summary, we make the following contributions:

1. We present *Latent Dependency Factor* (LDF), a new feature to learn the spatial and semantic dependencies within the combined source and target domains and close the gap between the two domains.

2. We introduce a novel two-stage autoencoder model to generate LDF. It learns

dependencies from the combined feature space of the clustered input data and the dependent variable.

3. We explore the settings for *spatial* transfer learning for $PM_{2.5}$ estimation in data-poor regions with similar feature space as the data-rich regions. This is a challenging problem consisting of untrained test locations and sparse target and source locations that causes minimal spatial autocorrelation.

4.2 Background

Previous studies have improved model predictions by imputing features from another dataset [118, 133] or generating synthetic samples to augment data [104, 209, 207]. The former leverages datasets with low marginal distribution, while the latter focuses on augmenting samples rather than features. In the domain of transfer learning, Daume et al. [41] and Duan et al. [53] introduce domain adaptation models — Feature Augmentation Method (FAM) and Heterogeneous Feature Augmentation (HFM), respectively — to create a common feature space using source and target features. These models are useful when the source and target domains have a dissimilar feature space, as noted by Pan et al. [166], whereas our approach incorporates spatial and semantic dependencies during ITL for domains with similar feature spaces, high marginal distribution, and low spatial autocorrelation.

4.3 Problem Formulation

Our problem comprises the source region with higher $PM_{2.5}$ sensors and the target region with fewer sensors. The data is heterogeneous due to diverse features and complex due to spatial and semantic dependencies between its samples.

Let X_f^S be the feature set for the source domain with m samples, and let X_f^T be

the feature set for the target domain with n samples, such that $m >> n$, and contains f features. Let Y^S and Y^T be the source and target domain labels ($PM_{2.5}$ levels). Hence, $D^S = (x_i^S, y_i^S)_{i=1}^m$ is the source domain dataset, where $x_i^S \in X_f^S$ is the feature vector for the i-th $PM_{2.5}$ monitor, and $y_i^S \in Y_S$ is the corresponding $PM_{2.5}$ value at the sensor. Similarly, $D^T = (x_i^T, y_i^T)_{i=1}^n$ is the target domain dataset with x_i^T and y_i^T representing i-th monitor and its $PM_{2.5}$ value, respectively.

Instance Transfer Learning (ITL) methodologies are employed when the two domains have varying marginal distributions. They find a reweighing function $w(x)$ that adjusts the importance of each sample in the source domain based on its relevance to the target domain. The importance weights $w(x_i^S)$ are calculated for each sample x_i^S in the source domain D^S, where $w(x_i^S)$ represents the degree of relevance of x_i^S to the target domain D^T. This degree of relevance is often calculated using probability densities, expressed as $w(x_i^S) = \frac{P_{D^T}(x_i^S)}{P_{D^S}(x_i^S)}$, where $P_{D^T}(x_i^S)$ and $P_{D^S}(x_i^S)$ is the probability density of x_i^S in the target domain and source domain respectively. The importance weights are applied to the source domain samples to obtain $\bar{D}_S = (\bar{x}_i^S, y_i^S)_{i=1}^m$ where $\bar{x}_i^S = w(x_i^S) \cdot x_i^S$. The reweighed source domain samples are used in the target domain for training; the combined domain is represented as $D^{\bar{S}T} = (x_i^{\bar{S}T}, y_i^{\bar{S}T})_{i=1}^{m+n}$.

Our goal is to improve the estimation of $PM_{2.5}$, such that the combined domain $D^{\bar{S}T}$ after reweighing source domain data D^S successfully captures the spatial and semantic dependencies.

4.4 Methodology

We introduce *Latent Dependency Factor* (LDF), a new feature imputed in the dataset to achieve *spatial* transfer learning for $PM_{2.5}$ estimation. The LDF has the following attributes: (1) It is highly correlated to the observed variable ($PM_{2.5}$ value), (2) It captures the spatial dependencies (spatial autocorrelation between nearby locations),

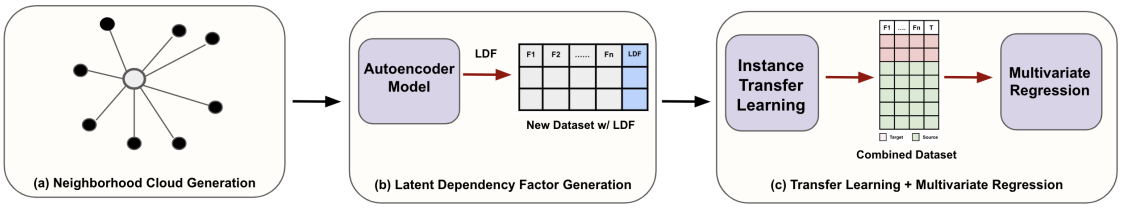

Figure 4.1: Framework for *spatial* transfer learning via *Latent Dependency Factor*

(3) It captures the semantic dependencies (semantic correlation in the combined data).

Imputing a new feature allows to learn a new loss function. Hence, if a function $f : X_f^{\bar{S}T} \rightarrow Y_T$ can predict the missing $PM_{2.5}$ values in the target domain D_T. Then, f is learned by minimizing the empirical risk as,

$$\min_f [\frac{1}{m+n} \sum_{i=1}^{m+n} \ell(y_i^{\bar{S}T}, f(x_i^{\bar{S}T})) + \lambda \cdot \Omega(f)] \tag{4.1}$$

where $\ell(y, \hat{y})$ is the loss calculated between true $PM_{2.5}$ value (y) and predicted value ($\hat{y}$) (here $f(x_i^{\bar{S}T})$), $\Omega(f)$ is a regularization term, and λ controls the trade-off between the empirical risk and model complexity. When a new feature is imputed, the empirical risk in (5.3) is transformed as,

$$\min_f [\frac{1}{m+n} \sum_{i=1}^{m+n} \widetilde{\ell}(y_i^{\bar{S}T}, \widetilde{f}(x_i^{\bar{S}T})) + \lambda \cdot \Omega(\widetilde{f})] \tag{4.2}$$

with the new trained regressor, $\widetilde{f}$ and loss function $\widetilde{\ell}$. *Hence, the new loss function allows obtaining a lower minimum.* The framework for *spatial* transfer learning via LDF contains 3 stages, as shown in Fig 4.1, which we elaborate further.

4.4.1 Neighborhood Cloud Generation

The first stage (Fig 4.1(a)) generates a neighborhood cloud of k similar data points for each sensor in the source and target regions. This cloud is training data for the two-stage autoencoder model, allowing each sensor to learn the spatial dependencies of its neighbors and semantic dependencies between the two domains. The similarity

between data points (sensors) is calculated by minimizing the $||L||_2$ distance across geographical, topographical, and meteorological features (see supplementary).

4.4.2 Generating Latent Dependency Factor (LDF)

After generating the neighborhood cloud, the subsequent steps involve generating the LDF, imputed as a new feature into the original dataset. This feature is derived using a two-stage autoencoder model (Fig. 4.2(a)), where the input dataset (neighborhood cloud) utilizes features – topographical, meteorological, geographical, and $PM_{2.5}$ levels. We believe these predictors influence the $PM_{2.5}$ levels at the objective location (centroid of the cluster). *E.g.*, given a sensor location, l_i, in the target region, the predictors such as the *wind-direction*, *elevation*, *population*, and more, for the surrounding sensors can influence the $PM_{2.5}$ levels at l_i (spatial autocorrelation). Additionally, the sensor location, l_i, can be semantically correlated to another location, l_j, in the source region, influencing the $PM_{2.5}$ levels at l_i in the combined dataset. In Fig. 4.2(a), each sensor has $(p+1)$ features with p features and a label. We first calculate the weight for each feature. This is achieved by finding the similarity (inverse distance) between the feature of the objective location and neighboring sensors. This allows sensors with influential features to be given more importance. Following the weighing, the features from m sensors are stacked together with the objective location to generate the input data of size $(m+1)\cdot(p+1)$. The $PM_{2.5}$ for the objective location is voided by setting it to 0. This high-dimensional data is summarized into the LDF, using the two-stage autoencoder model shown in Fig. 4.2(b).

Encoder-decoder Stage The *encoder-decoder* stage of the two-stage autoencoder model is similar to the standard autoencoder model, where the encoder first summarizes the input data to generate a latent value. The decoder employs backpropagation to train the autoencoder. The encoder and the decoder have three 1D-CNN layers with

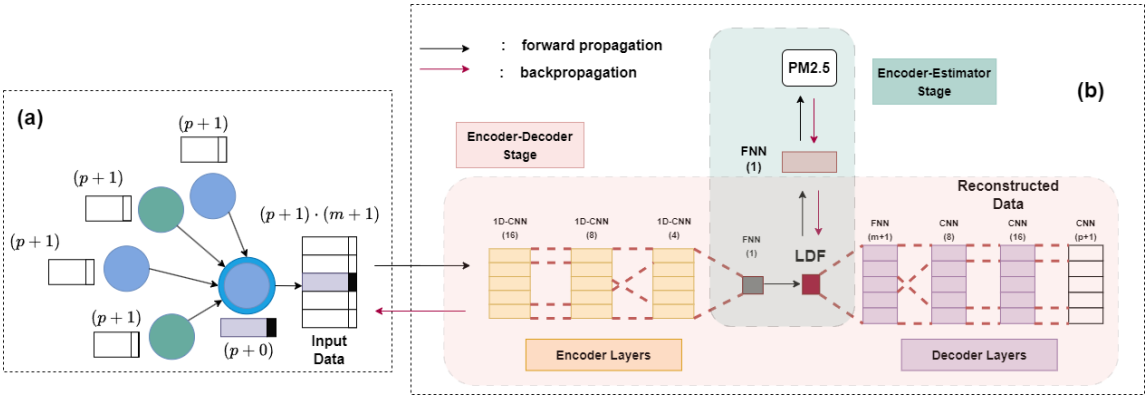

Figure 4.2: Two-stage autoencoder model for generating LDF.

varying filter sizes, as shown in Fig. 4.2(b). For the encoder, the kernel size of the first 2 CNN layers is chosen as 1 to achieve individual attention for each sensor and amplify the effectiveness of information summarization [112]. The third CNN layer has a kernel size 3 to retain the condensed pattern from multiple stations. Finally, the information is summed up using an FNN layer, which outputs the latent value, *i.e.*, the LDF value.

Encoder-estimator Stage Since the input data consists of multiple features, we increase the attention on $PM_{2.5}$ labels using the *encoder-estimator* stage. The estimator layer takes the encoded LDF value as input. It has a single FNN layer with a single weight and bias set. It utilizes back-propagation and $PM_{2.5}$ value of the objective location to train the encoder-decoder model and consequently optimize the LDF generation process. The autoencoder stages alternate training over the epochs. We also explore extending LDF to include Aerosol Optical Depth (AOD) [187] feature, which we call LDF-A and which measures the aerial density of aerosols such as smoke, dust, and PM particles, in the *encoder-esitmator* stage.

4.4.3 Transfer Learning and Multivariate Regression

In Fig.4.1(c), we employ Instance Transfer Learning (ITL) to mitigate discrepancies between source and target domain samples[71]. This involves reweighing the source domain samples to align them closer to the target domain. The reweighed source data is combined with the target data, creating a unified dataset reflecting both domains' characteristics.

This combined dataset is subsequently used to train a multivariate regressor for predicting $PM_{2.5}$ values. The choice of regressor can range as *polynomial-function* based, *decision-tree* based, or *ensemble* model. We employ an *ensemble* regressor for our framework, given their high prediction accuracy [49].

4.5 Evaluation

4.5.1 Datasets

We employ existing $PM_{2.5}$ dataset to perform transfer between varying climatic regions in the US [171]. In comparison to other datasets [33], this corpora draw from diverse sources (EPA, NLDAS-2, and NED for the US) and encompass a wide array of heterogeneous features such as *wind patterns*, *atmospheric pressure*, *humidity levels*, *potential energy* and more.

United States dataset.

As the US region has abundant $PM_{2.5}$ sensors, we select this dataset to simulate a transfer learning scenario within its geographical boundaries. The US dataset has daily averaged $PM_{2.5}$ levels for 2011 using 1081 sensors, as shown in Fig. 5.2, with over 249k samples and 77 features. Although the sample size should be 1081×365, some sensors were inactive on certain days (daily average active sensors: ~ 682). This contributes to missing temporal points in the dataset, which limits the application

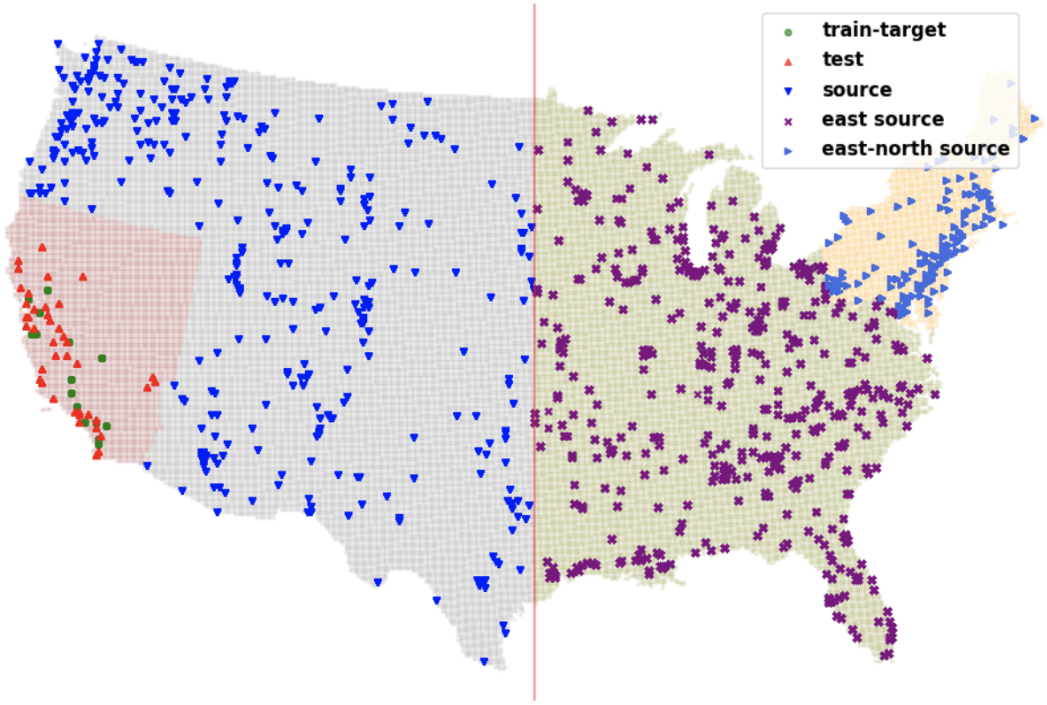

Figure 4.3: US $PM_{2.5}$ ground sensors. The points in the pink target region represent sample training (green) and testing (red) sensors. The green and yellow regions represent the eastern and north-eastern source regions, respectively.

of time-series forecasting methodologies. We follow the prior work [171] and use Layerwise Relevance Propagation [103] to extract 27 meteorological, topographical, and geographical features. As illustrated in Fig 5.2(a), we select two source regions, the eastern US (highlighted green; marker: $\mathbf{x}$) and north-eastern US (highlighted yellow; marker: $\blacktriangleright$) and a target region, California-Nevada (highlighted pink). Prior works [15] show that the California-Nevada region has a diverse landscape compared to the remaining US, thereby simulating a TL scenario with distribution shift and low spatial correlation among the two domains.

We sample the 128 target region sensors into sets of 5, 7, 9, and 11 sensors to have fewer samples. The remaining sensors are used for testing. For cross-validation (CV), we use 20 random samples per sensor. We extrapolate the active sensors per day and generate a neighborhood cloud for each sensor that includes both source

and target sensors. Next, the clustered data is used to generate the LDF which is fed to the transfer models. Our reported R^2 and RMSE values represent averages across the 20 CVs. The features are normalized before model training. For qualitative analysis (Section 4.5.5), we use $\sim$ 19.5 million unlabeled satellite data samples from the California-Nevada region.

4.5.2 Prediction Models

Machine Learning (ML) Models.

We select two popular ML models, **Random Forest Regressor** (RF) [99] and **Gradient Boosting Regressor**, trained on only the target region data and tested on the remaining test data. The RF and GBR have parameters varied as *n-estimators:* {100, 400, 1000}, *max-depth*: {4, 8, inf} with *max-leaf-node*: {4, 8, inf} for RF and *learning-rate*: {0.1, 0.5, 1.0} for GBR, to get the best fit.

The **Random Forest Regressor (RF)** is a popular supervised ensemble technique that builds multiple decision trees by bootstrapping samples and training a decision tree for each subset. Each decision tree, T_j within the forest, is grown by recursively partitioning the feature space to minimize the mean squared error (MSE) given by:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2,$$

where N is the number of samples, y_i is the original target label, and $\hat{y}_i$ is the predicted target label. The final prediction $\hat{y}$ for an input sample is obtained by averaging the predictions of all individual trees:

$$\hat{y} = \frac{1}{J}\sum_{j=1}^{J} T_j(x),$$

where J is the total number of trees and $T_j(x)$ represents the prediction of the j-th tree for input x.

Similarly, the **Gradient Boosting Regressor (GBR)**, is a supervised learner that utilizes iterative boosting to build an ensemble of weak learners, usaully decision trees. It achieves this by iteratively minimizing the residual error at each step. Given a target y and an initial prediction $\hat{y}^{(0)}$, each subsequent model $T^{(t)}(x)$ is trained on the residuals $r^{(t-1)} = y - \hat{y}^{(t-1)}$, where t represents the iteration step. The updated prediction at step t is given by:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \cdot T^{(t)}(x),$$

where η is the learning rate that represents the contribution of each tree to the final model. The goal of GBR is to reduce the loss (usually mse) over the iterations, by adjusting each learner's output.

Transfer Learning (TL) Models

We select competitive ITL models [136, 195, 100] for the regression task and train them on target and source region data. Below we elaborate on these models.

1. **Nearest Neighbor Weighing (NNW)**: The Nearest Neighbor Weighing (NNW) method [136] aims to reweight source samples based on their spatial similarity to target samples. In the NNW approach, each source sample serves as the center of a Voronoi cell. Mathematically, for a given source sample x_i^S in the source dataset $D^S = \{(x_i^S, y_i^S)\}_{i=1}^m$, a Voronoi tessellation is created around it, forming a region that includes all points in the feature space closer to x_i^S than to any other source sample. Formally, the Voronoi cell $V(x_i^S)$ associated with x_i^S is defined as:

$$V(x_i^S) = \{x \in \mathbb{R}^f : \|x - x_i^S\| \leq \|x - x_j^S\| \quad \forall j \neq i\},$$

where $\|\cdot\|$ denotes the Euclidean distance. By constructing Voronoi cells around

each source sample, we can quantify how well the source sample represents target samples in its vicinity.

Once the Voronoi cells are constructed, NNW assigns weights to each source sample based on the number of target samples falling within its cell. Let n_i represent the number of target samples $x_j^T \in X_f^T$ that fall inside the Voronoi cell $V(x_i^S)$ of the source sample x_i^S. The weight w_i for the source sample x_i^S is then proportional to n_i, allowing samples with more target neighbors to contribute more heavily to the model's training process. This reweighing is achieved by setting:

$$w_i = \frac{n_i}{\sum_{k=1}^{m} n_k},$$

where w_i is normalized across all source samples. The goal is to balance the representation of source samples that closely resemble target samples, thus improving model adaptability to the target domain.

The NNW method's effectiveness depends on the parameters used to define the neighborhood and the underlying regressor's capacity. In this study, we vary the number of neighbors in {6, 8, 10} and employ a Decision Tree Regressor as the base model with maximum tree depth set to {6, 8, inf} for optimal performance. These settings allow the model to adapt based on the structure of the data and the source-target alignment.

2. **Kullback–Leibler Importance Estimation Procedure (KLIEP)**: The Kullback–Leibler Importance Estimation Procedure (KLIEP) [195] is a method that reweights source samples by directly minimizing the Kullback-Leibler (KL) divergence between the source and target distributions. The KL divergence $D_{KL}(P_T \| P_S)$ quantifies the dissimilarity between the target distribution P_T and the source distribution P_S, with the goal of reweighting P_S such that it closely

approximates P_T. KLIEP accomplishes this by learning a set of importance weights $w(x)$ that minimize the KL divergence, formulated as:

$$D_{KL}(P_T \| P_S) = \int P_T(x) \log \frac{P_T(x)}{P_S(x)w(x)} \, dx.$$

Since $P_T(x)$ is generally unknown, KLIEP circumvents this by optimizing the weights $w(x)$ directly through a likelihood maximization approach, under the assumption that $w(x)$ makes $P_S(x)w(x) \approx P_T(x)$. This results in weights that enhance the representational similarity of the reweighted source distribution to the target distribution without estimating $P_T(x)$ explicitly.

To implement KLIEP, a kernel function $K(x, x')$ is used to map the feature space into a reproducing kernel Hilbert space, enabling flexible weighting based on the similarity between samples. We explore two types of kernels: radial basis function (RBF) and polynomial (poly) kernels, with parameters tuned to optimize performance. Specifically, the model parameters are varied as follows: the kernel type is chosen from {rbf, poly}, the γ parameter for the RBF kernel is varied across {0.1, 0.5, 1.0}, and we employ a Decision Tree Regressor as the base model with depth settings {6, 8, inf}.

In practice, the learned weights $w(x)$ are applied to each source sample x_i^S to reweight the training data, emphasizing samples from the source domain that align more closely with the target domain. This adaptive weighting helps to bridge the distributional gap between the domains, enhancing the model's performance on target data by leveraging the structural properties of KLIEP.

3. **Kernel Mean Matching (KMM)**: Kernel Mean Matching (KMM) [100] is a method that reweights source samples to align the mean of the source and target distributions in a Reproducing Kernel Hilbert Space (RKHS). The goal of KMM is to find a set of importance weights $w(x)$ that minimize the difference in

means between the reweighted source distribution P_S and the target distribution P_T in RKHS, represented as:

$$\left\| \frac{1}{m} \sum_{i=1}^{m} w(x_i^S)\phi(x_i^S) - \frac{1}{n} \sum_{j=1}^{n} \phi(x_j^T) \right\|,$$

where $\phi(x)$ denotes the feature mapping in RKHS, x_i^S and x_j^T are samples from the source and target domains, respectively, and $w(x_i^S)$ represents the importance weight for each source sample.

The objective is to minimize this difference in means while ensuring that the weights $w(x_i^S)$ do not deviate too far from 1 to prevent overemphasis on certain samples. To enforce this, KMM introduces constraints $0 \leq w(x_i^S) \leq B$ and $\sum_{i=1}^{m} w(x_i^S) = m$, where B is a regularization parameter controlling the extent of weighting adjustments. This optimization problem is solved using quadratic programming, leading to a set of weights that correct the distributional mismatch between P_S and P_T.

In practice, KMM employs a kernel function $K(x, x')$ to compute similarity in RKHS, which can be an RBF (radial basis function) or polynomial (poly) kernel. We vary the model parameters for optimal performance, selecting kernel types from {rbf, poly}, setting the γ parameter for RBF across {0.1, 0.5, 1.0}, and using a Decision Tree Regressor with depth options {6, 8, inf} as the base model. This adaptive reweighting of source samples improves the model's transferability to target data by reducing the mean discrepancy in RKHS.

4. **Fully-connected Neural Network (FNN)**: The FNN transfer model, although not an ITL model, is utilized to validate the performance of non-ITL models on the $PM_{2.5}$ data. It uses 3 fully connected layers: *nodes*: 128, *activation-function*: Relu, and 1 final layer with a single node and a linear activation function. It was trained on LDF-imputed source data and transferred by fine-tuning

over LDF-imputed target data.

The TL models are trained on data sans LDF, LDF, and LDF-A-imputed data. We use the GBR model as the multivariate regressor to predict $PM_{2.5}$, with parameters varied as: *estimators:* {100, 400, 1000}, *max-depth*: {4, 8, inf}, *max-leaf-node*: {4, 8, inf}, and *learning-rate*: {0.1, 0.5, 1.0} to get the best fit. The source code, datasets, and final hyperparameter values are available at:
https://github.com/YongbeeIngkle/spatial-transfer-learning.git.

4.5.3 Optimal k for Neighborhood Cloud

In Fig. 4.5(a), we use the eastern US as source data and vary the size of the neighborhood cloud (k) for the NNW [LDF] model as {4, 8, 12, 16}. Our choice of k mimicked optimizing parameters, ceasing at 16 due to high computational costs. We observe that $k = 4$ has the worst performance, while for the remaining values, there is no observable difference for sensors ≥ 9. For sensors ≤ 9, $k = 12$ has the most optimal performance. Hence, we chose $k = 12$ to optimize the computation and generalizability of the model.

4.5.4 Results and Analysis

In Table 4.1 and Table 4.2, we compare the performance of various models with the eastern US and the north-eastern US as source datasets, respectively.

Eastern US as Source Data. First, we compare the ML and TL sans LDF models. In Table 4.1, we observe that NNW, KLIEP, and KMM have a positive transfer (improved accuracy), with NNW having the best performance. We observe an unpredictable performance for the FNN transfer model, validating that non-ITL models are less suited for such transfer problems. Next, we illustrate the impact of the *Latent Dependency Factor* (LDF) on TL models. We observe an improvement in

Table 4.1: Source: Eastern US (best highlighted; second-best underlined)

Model	Sensors							
	5		7		9		11	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
RF	-0.082	8.855	0.002	8.565	0.066	8.387	0.071	8.311
GBR	-0.061	8.684	0.064	8.210	0.177	7.857	0.157	7.891
NNW	0.236	7.563	0.263	7.447	0.280	7.406	0.296	7.288
KLIEP	0.155	7.960	0.192	7.801	0.200	7.811	0.222	7.666
KMM	0.197	7.757	0.226	7.634	0.242	7.601	0.258	7.479
FNN	-0.064	8.818	-0.350	9.715	0.009	8.629	-0.039	8.765
NNW [LDF]	**0.247**	**7.494**	**0.336**	**7.061**	**0.378**	**6.874**	**0.378**	**6.838**
NNW [LDF-A]	0.225	7.596	0.298	7.230	0.359	6.973	0.359	6.924
KLIEP [LDF]	0.202	7.724	0.278	7.370	0.325	7.173	0.336	7.073
KLIEP [LDF-A]	0.232	7.584	0.267	7.427	0.319	7.201	0.330	7.100
KMM [LDF]	0.210	7.671	0.302	7.236	0.353	7.013	0.352	6.971
KMM [LDF-A]	0.196	7.723	0.295	7.277	0.330	7.134	0.333	7.067
FNN [LDF]	-0.255	9.532	-0.141	9.082	0.072	8.374	0.087	8.236
FNN [LDF-A]	-0.150	9.146	-0.105	8.990	0.091	8.275	0.078	8.287

estimation accuracy for NNW, KLIEP, and KMM (for both LDF and LDF-A), where NNW [LDF] is the best-performing model. For the FNN model, LDF has no notable effect as it caters to only ITL models. The high performance of NNW is due to the Voronoi tesselation neighborhood it uses for reweighing source samples. This allows it to capture similar samples in its neighbor, a spatially preferred reweighing technique.

North-eastern US as Source Data. In Table 4.2, we observe a positive transfer for NNW and KLIEP models, with NNW having the best performance. KMM shows a negative transfer [183] due to the high marginal distribution present between the target and source datasets [102]; unable to be minimized in reproducing kernel Hilbert space (RKHS) [100]. Like earlier, the FNN transfer model has an unpredictable performance. When the LDF is introduced, we observe an improvement in estimation accuracy for NNW and KLIEP models. NNW [LDF] and NNW [LDF-A] are the best-performing models. KMM [LDF-A] shows improvement for more sensors ($\geq$11). As expected, the FNN models using LDF and LDF-A show no improvement.

Table 4.2: Source: North Eastern US (best highlighted; second-best: underlined)

Model	Sensors							
	5		7		9		11	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
RF	-0.082	8.855	0.002	8.565	0.066	8.387	0.071	8.311
GBR	-0.061	8.684	0.064	8.210	0.177	7.857	0.157	7.891
NNW	0.199	7.732	0.294	7.286	0.301	7.297	0.298	7.257
KLIEP	0.098	8.180	0.219	7.650	0.263	7.494	0.270	7.408
KMM	-0.142	9.053	-0.070	8.809	0.232	7.640	0.246	7.526
FNN	0.022	8.448	-0.006	8.598	0.091	8.266	0.078	8.307
NNW [LDF]	**0.225**	**7.592**	0.317	7.157	0.376	6.886	**0.392**	**6.751**
NNW [LDF-A]	0.201	7.702	**0.320**	**7.122**	**0.378**	**6.873**	0.374	6.847
KLIEP [LDF]	0.164	7.889	0.275	7.363	0.353	7.011	0.360	6.924
KLIEP [LDF-A]	0.170	7.860	0.270	7.396	0.342	7.068	0.348	6.991
KMM [LDF]	-0.265	9.409	0.009	8.468	0.188	7.749	0.257	7.389
KMM [LDF-A]	-0.152	9.042	-0.029	8.566	0.172	7.845	0.288	7.260
FNN [LDF]	0.036	8.429	-0.052	8.761	0.131	8.061	0.237	7.566
FNN [LDF-A]	-0.060	8.774	0.045	8.390	0.159	7.983	0.207	7.708

4.5.5 Qualitative Analysis

While improving prediction accuracy is crucial, visualizing $PM_{2.5}$ patterns on geo-maps is also valuable. We visualize $PM_{2.5}$ estimations for the California-Nevada region and the Lima, Peru region in Fig. 4.4(a) and Fig. 4.4(b), respectively. For this analysis, we need a ground truth against which all the models can be compared. We use the GBR model, trained on all 128 monitors (249k+ samples) and estimated on the unlabeled satellite data ($\sim$ 19.5 M samples), and use its predicted geo-map as the assumed ground truth for verification. We use 9 sensors and the eastern US as source data for transfer models (NNW, NNW[LDF], NNW[LDF-A]).

Due to the scarcity of target domain data, this qualitative analysis aims to observe if transfer models successfully capture glaring $PM_{2.5}$ estimation patterns.

California-Nevada Region.

In Fig. 4.4(a), we observe that the NNW [LDF] model has the most accurate $PM_{2.5}$ estimation in the hotspots (solid boxes in the GBR map). It accurately captures

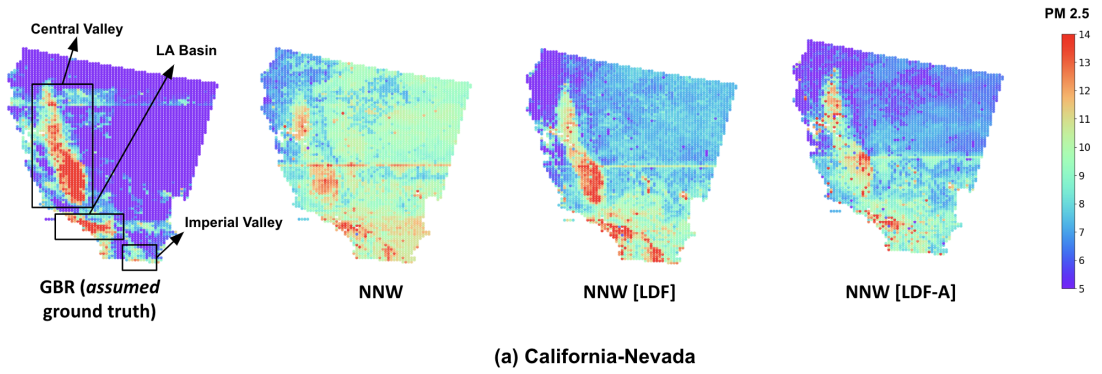

Figure 4.4: (a) Annual mean $PM_{2.5}$ prediction for *California-Nevada*, trained using GBR and NNW with and without LDF features (9 sensors). (b) Annual mean $PM_{2.5}$ prediction for Lima region trained using NNW models.

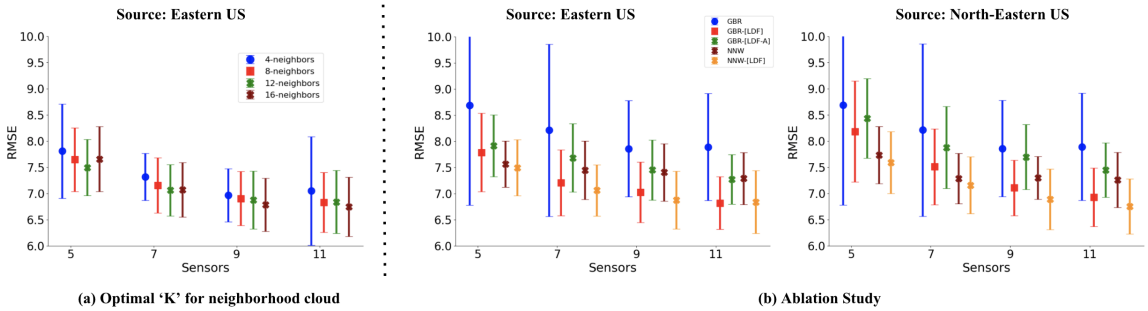

Figure 4.5: (a) Comparing performance of NNW [LDF] model when neighborhood cloud uses k = {4, 8, 12, 16} neighbors. (b) Ablation study comparing the performance of GBR, GBR [LDF], GBR [LDF-A], NNW, and NNW [LDF] models.

patterns in the *Central Valley* and the *Los Angeles Basin* but overestimates in the *Imperial Valley*. NNW [LDF-A] has the second-best performance but has a patchy estimation in the *Central Valley*. For NNW, we observe obscure patterns that are patchy and underestimated in the *Central Valley* and highly overestimated in the *Imperial Valley*.

4.5.6 Ablation Study

For the ablation study, we use GBR instead of ITL models to validate the performance of non-transfer models using LDF-imputed data. Fig.4.5(b) shows the comparison between GBR [LDF], GBR [LDF-A], GBR (target only), NNW, and NNW [LDF].

Table 4.3: Most correlated features (5) to $PM_{2.5}$ variable.

Method	**LDF**	**Pressfc**	**Dswrfsfc**	**Elev**	**Ugrd10m**
Corr Coeff	0.754	0.208	0.181	0.179	0.156

For both the eastern US and the north-eastern US as source data, GBR [LDF] is the second-best performing model. Though it doesn't outperform NNW [LDF], the improved predictions highlight LDF's effectiveness.

The performance of FNN [LDF] and FNN [LDF-A] in Table 4.1 and Table 4.2 further tests LDF with non-ITL models, confirming that LDF is effective with ITL and multivariate regression models but not other transfer models.

4.6 Discussion

While the evaluation results show the improvement using the LDF, we further analyze the correlation between LDF and PM2.5, as shown in Table 4.3, where LDF demonstrates the highest correlation with the dependent variable, indicating strong predictive power and feature importance [84]. This experiment uses an LDF-imputed dataset of 10 target sensors and eastern US source data.

4.6.1 Limitations and Future Work

While our methodology improves $PM_{2.5}$ estimation, further exploration, and alternate improvements are still needed, which we outline below.

Experiments with alternate datasets

Previous experiments with the US and Lima data are comprehensive but do not include datasets lacking spatial and semantic dependencies [33]. This was done primarily to ensure accurate and comprehensive data for modeling and estimation. Future plans include expanding our study to incorporate such datasets.

Capturing temporal trends

The LDF feature captures spatial and semantic dependencies but lacks focus on temporal trends in the data due to missing temporal points. In the future, we aim to extend this technique to time-series data, aiming for prediction rather than forecasting [252].

Extending to alternate domains

While our focus lies in $PM_{2.5}$ estimation, testing the LDF on alternate domains like wildfire estimation and weather forecasting is useful due to the presence of similar spatial patterns. Future studies should explore these applications and develop new LDF features accordingly.

4.7 Conclusion

This objective addresses the problem of *spatial* transfer learning for estimating $PM_{2.5}$ levels, emphasizing transfer between regions with low autocorrelation and predicting at unseen test locations. We aim to improve *instance transfer learning* (ITL) models, which often overlook spatial and semantic dependencies in the data. We introduce the *Latent Dependency Factor* (LDF) to capture these dependencies, integrating it as a new feature in both source and target datasets. Our experiments on US and Peru datasets demonstrate LDF's effectiveness in improving PM2.5 estimation. Furthermore, qualitative analysis of these datasets confirms that the LDF captures larger PM2.5 patterns missed by regular transfer models. While more future work remains in this space, we believe our approach of achieving *spatial* transfer learning using *Latent Dependency Factor* is a promising and novel solution for this highly complex domain.

Chapter 5

Transfer across regions w/ dissimilar feature space

This objective picks up from **RQ2** and focuses on transfer learning for spatiotemporal data especially for the pollution prediction use-case when the two domains do not share a feature space. The conditions are similar as the previous objective with irregularly spaced ground-sensors and missing temporal data (i.e. spatial and temporal irregularity). The overarching motivation of the problem also remains the same which involves estimating $PM_{2.5}$ for countries/regions with less ground-sensors. Additional conditions include, the source and target datasets belong to two spatially separated regions (i.e. countries) with varying data distributions as well as diverse meteorology and topography. Moreover, the regions are on opposite hemispherical ends and have different seasonality. This complicates the transfer process as seasonal similarity holds importance during the transfer process of air pollution estimation.

As previously mentioned – the two objectives, **(RQ2)** and **(RQ3)** were combined for a single paper published in the conference ECML-PKDD 2024 [81]. We have extended Lima experiments and also plan to include transfer experiments on other countries to be compiled as a journal extension of our paper.

5.1 Rationale

As discussed in the previous chapter, the installation of $PM_{2.5}$ ground sensors yield highly accurate data as these sensors employ intricate measurements techniques like gravimetric analysis [7] and more. However, scaling, installation and maintenance costs [118] is high for such ground sensors [47], that creates an imbalance of *data-rich* (developed) and *data-poor* (developing) regions based on access to such resources for $PM_{2.5}$ data collection and estimation.

Hence, often the *data-poor* regions with sparse $PM_{2.5}$ ground-sensor networks, rely on alternate data collection techniques like satellite-based measurements for air pollution estimation. However, the satellite data lacks accuracy and granularity compared to the ground-sensors. Moreover, their performance highly varies across geographic locations i.e. the variations caused due to changes in topographical and meteorological conditions also affect the data curation process. For example, the $PM_{2.5}$ dataset has $PM_{2.5}$ measurements using ground sensors as well as factors affecting these measurements such as temperature, pressure, wind directions, and more. It might be the case that the curation process is not consistent across the globe with certain features like forest cover, highway roads, and more missing in the $PM_{2.5}$ datasets for certain regions. This creates data non-uniformity or feature variations between two geographically different regions.

Moreover, the spatiotemporal dynamics of $PM_{2.5}$ that is affected by seasonal variations complicate the transfer learning process if the two regions do not have the same seasonality. For example, the *data-rich* region A might witness summers during the traditional June to August compared to *data-poor* region B that can witness summers during December to February. While $PM_{2.5}$ occurrence is not seasonal [238], a varying seasonality between regions add more complexity during the training process as the model associates meteorological trends with certain periodicity. Hence, the seasonal patterns found in data-rich regions in one hemisphere may not apply directly

to data-poor regions in another as an effect of these variations. Hence, in this objective, we focus on transfer across regions with dissimilar feature space with the inclusion seasonality focused experiments.

Utilizing the **RQ2** transfer model framework that uses Latent Dependency Factor as an additional feature, this objective incorporates solution to some previous and some newer problems as defined below:

1. We employ the *Latent Dependency Factor* (LDF) feature that learns spatial and semantic dependencies within the combined source and target domains and consequently helps close the gap between the two domains. To generate the LDF feature, we utilize the two-stage autoencoder model introduced previously that learns dependencies from the combined feature space of the source and target domain data.

2. We explore the settings for *spatial* transfer learning for $PM_{2.5}$ estimation in data-poor regions with the target and source domains having a dissimilar feature space. This is a challenging problem as it requires feature standardization, and seasonality focused experiments. We elaborate these seasonality experiments using two scenarios employing seasonality-agnosticism and seasonality-matching for accurate model prediction.

3. We deploy our technique in Lima, Peru, and validate the results by domain experts due to the scarcity of true labels. This offers insights into the real-world application of our technique and its effectiveness.

5.2 Problem Formulation

Our problem comprises a source region with a higher density of $PM_{2.5}$ sensors and a target region with a lower density of sensors. The data is heterogeneous due to

distinct feature space for the two domains and complex due to spatial and semantic dependencies between its samples. As introduced in the previous chapter, X_{f1}^S with m samples and $f1$ features, represents the source domain and X_{f2}^T with n samples and $f2$ features, represent the target domain. Initially, the feature spaces in the source and target domains differ; however, a standardization process is applied to handpick f similar features from each domain to create a partially aligned feature space. This results in standardized feature sets $\tilde{X}_f^S \subset X_{f1}^S$ and $\tilde{X}_f^T \subset X_{f2}^T$, where f features are shared between domains. Hence, if Y^S and Y^T be the $PM_{2.5}$ levels (labels) for the source and target domains, then source domain dataset is defined as $D^S = \{(x_i^S, y_i^S)\}_{i=1}^m$, and the target domain dataset is defined as $D^T = \{(x_i^T, y_i^T)\}_{i=1}^n$ (also explained in the previous chapter).

Instance Transfer Learning (ITL) methodology is useful when the source and target domain data has differing marginal distributions. The weight of source samples is represented as $P_{D^S}(x_i^S)$, where $P_{D^T}(x_i^S)$ and $P_{D^S}(x_i^S)$ denote the density of x_i^S in the target and source domains, respectively. Consequently these weights are applied to the source domain to yield $\bar{D}_S = \{(\bar{x}_i^S, y_i^S)\}_{i=1}^m$, where $\bar{x}_i^S = w(x_i^S) \cdot x_i^S$. The reweighed source samples are then combined with target data to form the combined domain as, $D^{\bar{S}T} = \{(x_i^{\bar{S}T}, y_i^{\bar{S}T})\}_{i=1}^{m+n}$.

Our objective is to improve $PM_{2.5}$ estimation such that the LDF feature captures spatial and semantic dependencies as well as overcomes the complexity due to reduced feature space and seasonal differences.

5.3 Methodology

The *Latent Dependency Factor* (LDF), imputed in the dataset, improves estimation by capturing spatial dependencies (spatial autocorrelation between nearby locations) and semantic dependencies (semantic correlation in the combined data) present in

the two datasets. In the previous chapter, we proved that imputing a new feature allows to learn a new loss function, thereby, minimizing the empirical risk that can be represented as. $\min_f[\frac{1}{m+n}\sum_{i=1}^{m+n}\ell(y_i^{\bar{S}T}, f(x_i^{\bar{S}T})) + \lambda \cdot \Omega(f)]$ before LDF is introduced and as, $\min_f[\frac{1}{m+n}\sum_{i=1}^{m+n}\widetilde{\ell}(y_i^{\bar{S}T}, \widetilde{f}(x_i^{\bar{S}T})) + \lambda \cdot \Omega(\widetilde{f})]$ after LDF is introduced. The framework for *spatial* transfer learning via LDF contains 3 stages as elaborated next.

5.3.1 Neighborhood Cloud Generation (Input Dataset)

The first stage generates the neighborhood cloud of k similar data points for each sensor in the source and target regions where the similarity is calculated by minimizing the $||L||_2$ distance across the features. This cloud is utilized as the training data for the two-stage autoencoder model, allowing each sensor to learn the spatial dependencies of its neighbors and semantic dependencies existing between the two domains.

5.3.2 Generating LDF via Two-stage Autoencoder Model

The second stage generates the LDF and integrates it as a new feature into the original dataset. The LDF is computed using a two-stage autoencoder model shown in Fig. 5.1. Each sensor has $p+1$ features, which include p predictors and a label. Feature weights are assigned to all features with respect to their similarity, measured by inverse distance between the objective location and neighboring sensors, favoring the selection of influential features. Data from m neighboring sensors and the objective location are stacked into a matrix of size $(m+1)\cdot(p+1)$, with the $PM_{2.5}$ value for the target location initialized as 0. This high-dimensional matrix is reduced to the LDF through the two-stage autoencoder shown in Fig. 5.1(b).

Stage 1: Encoder-decoder: The *encoder-decoder* stage of two-stage autoencoder works similarly as the standard autoencoder in which encoder compresses input into latent representation, while the decoder performs training by backpropagation. Each

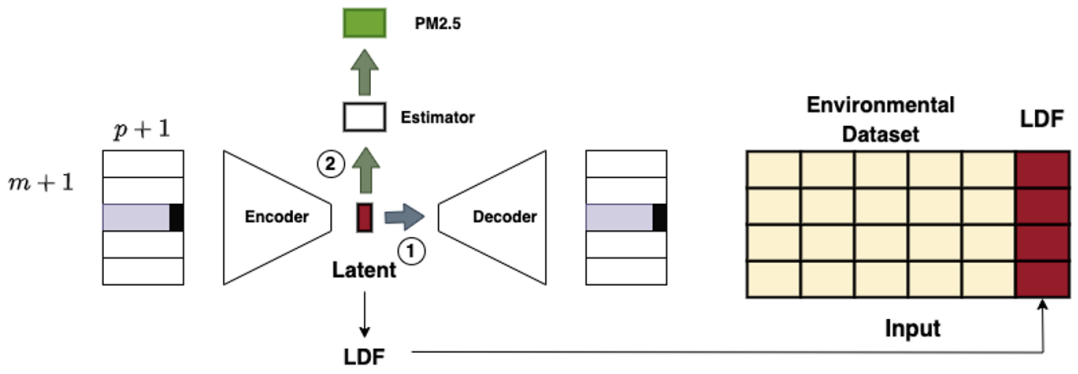

Figure 5.1: Two-stage autoencoder model for generating LDF.

of the Encoder/Decoder architectures consists of three 1D-CNN layers with the first layers having a kernel of size 1, and the third layer having a kernel size of 3. An FNN layer aggregates all the processed information and generates the LDF.

Stage 2: Encoder-estimator: To increase attention on the PM2.5 labels, the encoder-estimator stage includes only 2 FNN-based estimator layer, which takes the encoded LDF as input. It utilizes backpropagation and the PM2.5 value of the target location to further fine-tune the generation of LDF. Additionally, the extension of LDF with the Aerosol Optical Depth(AOD) [187] is referred to as LDF-A and is integrated into the encoder-estimator stage.

5.3.3 Regression Transfer Learning

In the last stage, the ITL methodology is used to mitigate the gap between the source and target domain samples by reweighing source samples [71]. Subsequently, the reweighed source data is combined with the target data into a single dataset and used to train a multivariate regressor that predicts the $PM_{2.5}$ values. In this framework, an ensemble regressor is used because it outperforms other methods in predictive performance [49].

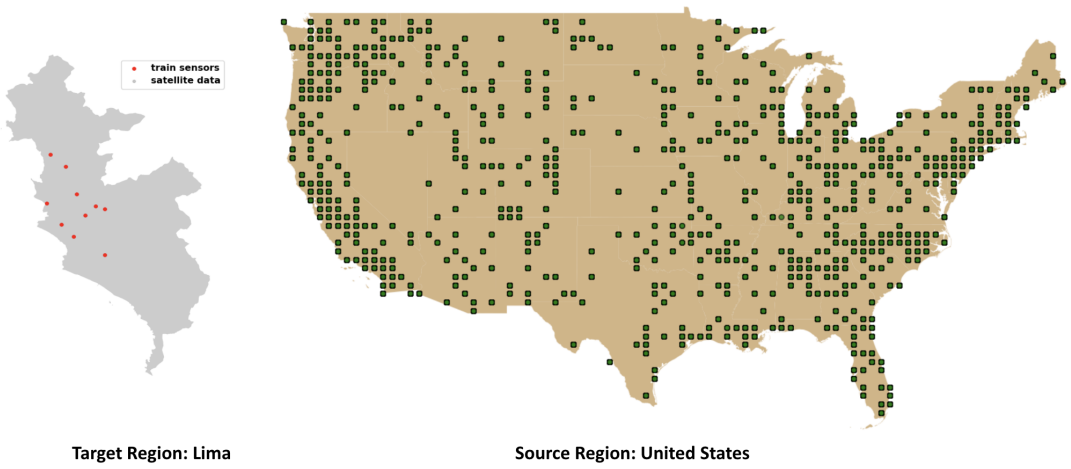

Figure 5.2: Lima and United States $PM_{2.5}$ ground sensors. For the Lima region –the red points represent ground sensors and the gray region indicates (unlabeled) satellite measurements. For the US region – the green points represent the ground sensors.

5.4 Evaluation

5.4.1 Datasets

We employ $PM_{2.5}$ data from the Lima city in Peru [210]. This dataset is created from sources such as SENAMHI and JHU and similar to the US $PM_{2.5}$ dataset, it also consists of heterogeneous features such as *wind patterns*, *atmospheric pressure*, and more. We utilize the US $PM_{2.5}$ dataset [171] as the source domain dataset, derived from sources – EPA, NLDAS-2, and NED (more details in the previous objective).

Lima dataset

Given the dearth of sensors in the Lima data, it is a use case of real-world transfer learning where the source data is the complete US dataset (249k+ samples, 27 features). Lima region has 10 $PM_{2.5}$ sensors, as shown in Fig 5.2(b), with 2419 samples and 21 features for the year 2016. Lima and the US datasets have only 14 common features. For the qualitative analysis, the Lima satellite data contains 5959 samples covering the entire Lima region as shown in Fig 5.2(b). We use all 10 sensors and the US

dataset to construct the neighborhood cloud data. By aligning each *day of the year (doy)* between the two datasets (e.g., day 17 in Lima matched with day 17 in the US), we extrapolate sensors for the day and generate the clusters. The doy matching is deliberately not for the same year or season to have a real-world transfer condition with minimal alignment.

United States dataset

Given the abundance of $PM_{2.5}$ sensors in the US, it serves as the source dataset. The dataset consists of daily $PM_{2.5}$ averages from 2011, recorded by 1,081 sensors, amounting to more than 249,000 samples with 77 features. Following previous works [171], LRP [103] is applied to extract 27 most informative meteorological, topographical, and geographical features.

5.4.2 Prediction Models

Although only NNW results are shown, we revisit the transfer learning and machine learning models used in our experimentation which are detailed in **RQ2**. For machine learning models, we choose Random Forest Regressor (RF) and Gradient Boosting Regressor (GBR). RF and GBR were trained using target region data and tested with the remaining sensor data. The hyperparameters were varied as follows: *n-estimators* (100, 400, 1000), *max-depth* (4, 8, inf), *max-leaf-node* (4, 8, inf) for RF, and *learning-rate* (0.1, 0.5, 1.0) for GBR to achieve the best fit.

For transfer learning models, NNW reweighs source samples with Voronoi tessellation, with parameters varied as: *neighbors* ({6, 8, 10}), and a Decision Tree Regressor with *depth* ({6, 8, inf}). The remaining transfer models and their hyperparameters can be referred in the previous chapter. We provide some insights into how they work. The goal of KLIEP is to reweigh the samples such that the KL divergence between source and target domains is minimized. KMM aligns source and target domain data

means in a reproducing kernel Hilbert space for similar parameters. FNN consists of three fully connected layers consisting of 128 nodes, using the Relu activation function, followed by one node and linear activation, trained on LDF-imputed source data, and fine-tuning on target data.

Three transfer learning models - without LDF, with LDF, and LDF-A, were trained. GBR was used as the multivariate regressor for $PM_{2.5}$ prediction, while the parameters *estimators* (100, 400, 1000), *max-depth* (4, 8, inf), *max-leaf-node* (4, 8, inf), and *learning-rate* (0.1, 0.5, 1.0). Source code, datasets, and final hyperparameter values are provided in the previous objective.

5.4.3 Optimal k for Neighborhood Cloud

In the previous objective we expanded upon the choice of 'k' for the neighborhood cloud in Fig. 4.5(a) where we use the eastern US as source data and vary the size of the neighborhood cloud (k) for the NNW [LDF] model as $\{4, 8, 12, 16\}$. We observe that sensors ≤ 9, $k = 12$ has the most optimal performance. Therefore, we chose $k = 12$ to optimize the computation and generalizability of the model.

5.4.4 Feature Standardization

We match the features of the Lima and United States dataset based on commonalities between the two (as shown in Table 5.1). We utilize the domain knowledge and excise features that do not match as well as are not measured on the same scale. This allows us to match 14 features from a total of 27 and 21 features for the United States and Lima respectively. The meteorological feature set consists of co-variates representing temperature, pressure, humidity, surface radiation, and wind direction. Whereas the topographical feature set consists of co-variates representing elevation, and population density. It should be noted that having less features affects the prediction performance which the LDF model is able to overcome as shown in the results section below.

Lima	Description	United States
doy	Day of year	day
month	Month	month
Lon	Longitude	Lon
Lat	Latitude	Lat
temp_2m	Temperature at 2m	nldas_tmp2m
rhum	Relative Humidity	nldas_rh2m
surf_pres	Surface Pressure	nldas_pressfc
conv_prec	Precipitation	nldas_pcpsfc
short_radi_surf	Short wave surface radiation	nldas_dswrfsfc
DEM	Elevation	elev
Population	Population per Km	pd
zonal_wind_10m	U-wind	nldas_ugrd10m
merid_wind_10m	V-wind	nldas_vgrd10m
AOD550	Modis AOD 550	gc-aod

Table 5.1: Matching the features of Lima and United States dataset

5.4.5 Results and Analysis

To analyze Lima results, we perform qualitative analysis, primarily due to the dearth of labels for the satellite measurements. We visualize the $PM_{2.5}$ patterns on geo-maps. We utilize $PM_{2.5}$ dataset for the US as the source region data whereas the Lima, Peru dataset with 10 stations is used as the target region data. Since the true labels for Lima were unavailable, we consulted the domain experts (environmental scientists) for the analysis. Due to the scarcity of target domain data, this qualitative analysis aims to observe if transfer models successfully capture glaring $PM_{2.5}$ estimation patterns for standardized feature space and differing seasonality between the two domains. In the previous objective, we identified the ITL model, Nearest Neighbor Weighing (NNW) to have the most optimal performance, therefore, we localize our qualitative experiments using just the NNW model. The insights from domain experts measure the 'goodness' or accuracy of the NNW models namely – without LDF, with LDF and LDF-A. Additionally, we tried two of experiments to validate seasonal performance of our methodology. The first experiment didn't perform seasonal matching between the source (US) and target (Lima) datasets and estimated daily average seasonal

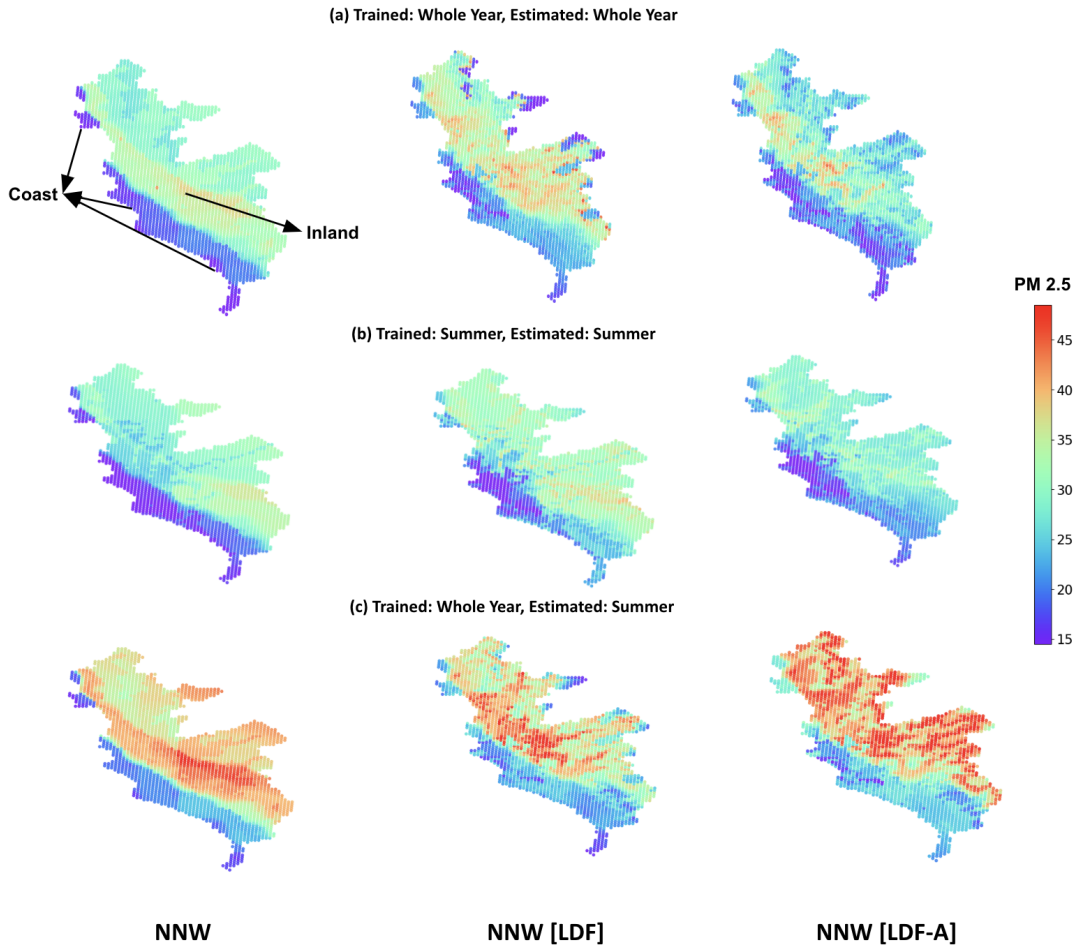

Figure 5.3: (a) Annual mean $PM_{2.5}$ prediction for *Lima*, trained on the whole year data without seasonal matching. (b) Seasonal mean $PM_{2.5}$ prediction for Lima region trained using summer season matching. (c) Seasonal mean $PM_{2.5}$ prediction for *Lima*, trained on the whole year data without seasonal matching.

and annual $PM_{2.5}$ values (as shown in Figure 5.3(a) and Figure 5.3 (c)). The second experiment performed seasonal matching by extrapolating summer seasons (June to August for US) (December to Feburary for Lima) for both the domains and then matching each day to generate the neighborhood cloud dataset.

Seasonality Focused Experiments for the Lima Region

Scenario 1: In Figure 5.3, we plot qualitative results with and without seasonal matching between the source and target domains. For Figure 5.3(a), we utilize the

entire year data from source (US) and target (Lima) domain and match each day to generate the neighborhood cloud for each sensor. We observe that all models exhibit lower $PM_{2.5}$ levels near the *coast* and higher levels moving *inland*, a pattern validated by domain experts. However, NNW [LDF] has a clearer concentration gradient of *inland* $PM_{2.5}$ compared to the other models. Near the *Andes mountain ranges*, the $PM_{2.5}$ is the lowest, which the NNW[LDF] model accurately captures but slightly and highly overestimated by the NNW [LDF-A] and NNW models, respectively. These observations confirm the improvement of prediction by LDF-based TL models.

Scenario 2: For Figure 5.3(b), we utilize the summer season data from source (US) (from June to August) and target (Lima) (from December to Feburary) domains and match each day of the 90 days window to generate the neighborhood cloud for each sensor. We observe similar trends as Figure 5.3(a) with lower $PM_{2.5}$ near *coast* and higher values moving *inland*. Additionally, the concentration gradient for NNW with LDF features is better compared to the NNW model. Among the NNW[LDF] and NNW[LDF-A], the high inland $PM_{2.5}$ is accurately captured by NNW[LDF] whereas the $PM_{2.5}$ near the *Andes mountain ranges* is slightly lower for NNW[LDF-A]. Hence, the NNW models with LDF imputed features capture the $PM_{2.5}$ much accurately compared to NNW where NNW[LDF] has the most optimal performance.

Scenario 3: For Figure 5.3(c), we utilize the whole year data from source (US) and target (Lima) domains and generate the neighborhood cloud for each sensor to predict the summer season $PM_{2.5}$ values. This helps validate the estimation of Figure 5.3(b). We observe a shift in $PM_{2.5}$ estimation with all models predicting higher concentration gradient during the summer. While NNW model still doesn't capture the correct concentration gradient, the NNW[LDF-A] fills the entire map with high $PM_{2.5}$ values. In hindsight, both models highly overestimate $PM_{2.5}$ values in all three regions – *coast*, *inland* and *Andes mountain ranges*. Additionally, the concentration gradient

for NNW[LDF] and NNW[LDF-A] models is better compared to the NNW model. Among the LDF imputed models, NNW[LDF] manages to capture high *inland* $PM_{2.5}$ with lower values in the remaining regions. The NNW[LDF] values are the closest to the real-world $PM_{2.5}$ estimates during summer in Lima. It should also be noted that these estimations are still not completely accurate and this qualitative analysis attempts to shed a light on how these models capture the large glaring patterns.

Additional Experiments: We performed a 60:40 train-test split on the Lima sensors and trained NNW and NNW [LDF] TL models using 3-fold cross-validation, with the complete US as the source data; trained and estimated on the entire year. The results for [R^2, RMSE] for NNW and NNW[LDF], respectively, were [0.476, 9.852] and [0.558, 9.091]. Hence, NNW [LDF] outperforms NNW, thereby validating the qualitative analysis results.

5.5 Discussion, Limitations and Future Work

Despite the lack of ground labels for deploying the LDF-based NNW model in Lima, it is important to address the pressing issue: Lima is the second most polluted city in the Americas [202] and suffers from a scarcity of sensors [210] (since Peru is a developing country). Our model provides a groundbreaking outcome in $PM_{2.5}$ estimation for Lima and serves as a vital first step toward implementing similar models in other *data-poor* regions. We believe our methodology has room for improvement in terms of expanding experiments, baselines and ablations. We outline these below:

Experiments with alternate (*data-poor* region) datasets

While our experiments with the US and Lima data provides qualitative analysis for the LDF model estimation, however, we need to extensively quantify transfer between countries with dissimilar features. Our future experiments involve utilizing

altenrate datasets belonging to countries – India, Taiwan, and more for a more robust performance measure.

While this future work has previously been covered, but we plan to also include transfer across countries with dissimilar feature space for open-source datasets. Since, these datasets often contains very few meteorological and topographical features affecting the $PM_{2.5}$ values, it would be highly interesting to see the prediction performance of LDF framework for them.

Extending to alternate irregular spatiotemporal domains

As mentioned in the previous chapter, we would like to test the LDF model on alternate domains like wildfire estimation and weather forecasting, especially the domain that utilize sensor measurements and contains irregularities. Hence, the future studies should explore these applications and develop new LDF features accordingly.

5.6 Conclusion

This objective addresses the problem of transfer learning for irregular spatiotemporal data with dissimilar feature space. We utilize the LDF framework to estimate $PM_{2.5}$ levels where the transfer takes place between regions with low autocorrelation and predicting at unseen test locations. We aim to improve *instance transfer learning* (ITL) models, which often overlook spatial and semantic dependencies in the data. We first standardize the feature space between the two datasets – US and Lima. We then perform two types of seasonality focused modeling – with and without seasonal matching between the source and target regions. Our results show that seasonal matching is important for prediction over a shorter time range whereas for a longer duration (annual estimation), no seasonal matching is required. The results also validate the consistent performance of the LDF model with or without

seasonal matching. In conclusion, the model performs well for long-range prediction without seasonal matching but for shorter range, it requires such an explicit matching. Moreover, our qualitative experiments on US and Peru datasets demonstrate LDF's effectiveness in improving PM2.5 estimation and capturing larger PM2.5 patterns missed by regular transfer models. While more future work remains in this space, we believe our approach of achieving *spatial* transfer learning using *Latent Dependency Factor* is a promising and novel solution for this highly complex domain and should be explored further.

Chapter 6

Future Work

Previous suite of transfer models that I designed focus on instance transfer learning techniques that involves reweighing source domain samples. This reweighing is also known as domain adaptation. Since only one source dataset was utilized, such reweighing can also be called single source domain adaptation (SSDA). A logical next step to SSDA is to effectively apply domain adaption with multiple source domain dataset i.e. multi-source domain adaptation (MSDA). Moreover, while classical and neural regression models are useful for continuous datasets, they are also parametric i.e. with fixed weights and parameters. This begs the question about non-parametric approaches that can adapt their parameters based on data complexity given the non-iid nature of the spatiotemporal data. In the following sections we explore these concepts more and provide a direction for future research.

Multi-Source Domain Adaptation (MSDA)

6.1 Introduction

Multi-source domain adaptation (MSDA) paradigm involves knowledge transfer from multiple labeled source domains to an unlabeled or sparsely labeled target domain [145,

52, 69]. In contrast to single-source domain adaptation (SSDA), MSDA learns from diverse source domains with distinct data distributions, and consequently generalizes on the target domain. The key challenge in this scenario is due to the heterogeneity of various source domain distributions and their dissimilarity to the target domain distribution. Traditional machine learning models fail in scenarios where the train and test data distribution are dissimilar. This also falls within the i.i.d (independent and identically distributed) assumption which presumes the features across both domain have similar independent distributions. However, the non-i.i.d characteristics in the data causes the dissimilarity between the train and test domains. This can be extended to the case of transfer learning where target data has dissimilar distribution to the source data. Therefore, an optimal MSDA considers both marginal and conditional probability differences i.e. differences between the feature spaces as well as the differences between label given features [196].

For the use-case of irregular spatiotemporal data, source domains can be data from multiple regions (with differing $PM_{2.5}$ values for each region). Training on multiple domains allows for an improved learning of invariant representation from each domain. Hence, an MSDA model trained on multiple sources can be utilized to make predictions (eg., $PM_{2.5}$ levels) in a new region (target). This approach can be counterintuitive (leads to negative transfer) if naive aggregation over multiple source datasets is involved. An independent learning over each source domain distribution leads to a suboptimal performance as no single source fully captures the variability of the target distribution. Therefore, we discuss an optimal solution for multi-source domain adaptation below as also presented by Mansour et al. [145].

6.2 MSDA via Weighted Combination

Mansour et al. [145] propose a MSDA solution where the target distribution, $D_T(x)$ is considered as a mixture of (k) source distributions, $D_i(x)$, and hence modeled as:

$$D_T(x) = \sum_{i=1}^{k} \lambda_i D_i(x),$$

with λ_i represents the weight (contribution) of each source D_i to the target. The weights are constrained to the simplex $\Delta = \{\lambda : \lambda_i \geq 0, \sum_{i=1}^{k} \lambda_i = 1\}$. Hence, the goal is to combine hypotheses $h_i(x)$, trained on the source distributions, into a single hypothesis $h(x)$ for the target domain. There are two hypotheses combination rules for the MSDA scenarios:

1. **Linear Rule**:

$$h(x) = \sum_{i=1}^{k} \lambda_i h_i(x).$$

   The linear rule provides equal weights to each source data distribution. Given its simplicity, it fails when the distributions are varying.

2. **Distribution-weighted Rule**:

$$h(x) = \frac{\sum_{i=1}^{k} \lambda_i D_i(x) h_i(x)}{\sum_{j=1}^{k} \lambda_j D_j(x)}.$$

   The distribution-weighted rule weighs each source hypothesis h_i based on the local density $D_i(x)$ of the corresponding source distribution for input x. The rule ensures that hypotheses from source distributions with higher local densities contribute more to the target distribution. The combined prediction is normalized as, $\sum_{j=1}^{k} \lambda_j D_j(x)$, such that weights of all sources equals 1.

6.2.1 Learning Optimal Weights for Distributed Weighing

To determine the source distribution weights λ_i, the problem is formulated as an optimization task minimizing the loss L, which is defined as, $L(D_T, h, f)$ over the target distribution. Hence, we minimize L as,

$$\min_{\lambda \in \Delta} \sum_{i=1}^{k} \lambda_i L(D_i, h_i, f),$$

where f is the true target function, and L is a loss function (e.g., mse). This convex optimization ensures that the combined hypothesis $h(x)$ minimizes the target distribution while also considering source contributions. In practice, Maximum Mean Discrepancy (MMD) is used to measure the similarity between the weighted source distributions and the target, facilitating iterative updates of λ. Additionally, the distribution-weighted rule can handle non-linearity for source-target alignment using the density $D_i(x)$. A regularization term, $\|\lambda\|_2$, is added for noisy or sparse data.

6.3 MSDA: Two-Stage Domain Adaptation

Sun et al. [196] present a two-stage domain adaptation approach that reduces the gap between source and target domains by sequentially reducing the discrepancies in the marginal and conditional probabilities. In context of domain adaptation, the *marginal probability* $P(x)$ represents feature distribution for the domains, while the *conditional probability* $P(y|x)$ represents the relationship between features and labels. Classical domain adaptation models reduce the difference in marginal probabilities, $P(x)$ as they can vary due to differences in underlying domain distributions. However, the domain conditional probabilities, $P(y|x)$ can also observe shift based on how features map to labels. Hence, a two-stage framework allows to mitigates these domain distribution challenges . We elaborate on the two stages below:

6.3.1 Stage 1: Marginal Probability Alignment

The first stage of the two-stage domain adaptation approach aligns the marginal probabilities $P(x)$ between the source and target domain distribution. This alignment is due to the variance between the feature distributions $P(x_s)$ and $P(x_t)$ of the two domains. The source samples are reweighed to be similar to the target samples. Techniques such as Maximum Mean Discrepancy (MMD), Kullback-Leibler (KL) divergence, Importance Sampling, and more can be utilized for source sample reweighing.

Let $D^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ represent the labeled source domain data, $D_u^t = \{x_i^t\}_{i=1}^{n_t}$ the unlabeled target domain data, $P_s(x)$ and $P_t(x)$ the marginal distributions of the source and target domains, and $P_s(y|x)$ and $P_t(y|x)$ their respective conditional distributions. To align the marginal probabilities, source data weights $\alpha_s(x_i^s)$ are adjusted to minimize the discrepancy:

$$\min_{\alpha_s} \mathcal{D}_M(P_s(x), P_t(x)),$$

where $\mathcal{D}_M$ represents a divergence measure (e.g., Maximum Mean Discrepancy (MMD) or Kullback-Leibler (KL) divergence). The re-weighted source data becomes:

$$\{(\alpha_s(x_i^s)x_i^s, y_i^s)\}_{i=1}^{n_s}.$$

This reweighing allows the source samples to align closely to the target samples, thereby laying the foundation for the next stage of adaptation.

6.3.2 Stage 2: Conditional Probability Alignment

The second stage of the domain adaptation approach focuses on dissimilar conditional probabilities, $P(y|x)$ of the source and target domain distributions. Following the alignment of marginal probability alignment, we train multiple hypotheses, h_s

(equivalent to the number of source domains) on the reweighed source data. These hypotheses then estimate the target labels and the estimations are combined as a matrix, H_S. Hence, H_S represents the matrix for estimated target labels, where each column corresponds to a source hypothesis. Then H_S can be represented as,

$$H_S = \begin{bmatrix} h_1(x_1^t) & h_2(x_1^t) & \dots & h_k(x_1^t) \\ h_1(x_2^t) & h_2(x_2^t) & \dots & h_k(x_2^t) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_{n_t}^t) & h_2(x_{n_t}^t) & \dots & h_k(x_{n_t}^t) \end{bmatrix}.$$

The weights for source domain hypotheses h_s are estimated using the smoothness assumption which posits that if two points in the feature space are similar (or close), their corresponding labels should also be similar. Hence, the weight for the various source domains is found by minimizing the difference in predicted labels between similar target samples.

6.4 MSDA: Adversarial Learning

The goal of multi-source domain adaptation using adversarial learning [249] is to generate domain-invariant representations to bridge the distribution difference between (multiple) source and target domain. Hence, these representation have two characteristics:

1. They are *domain-invariant*, i.e. their origin domain (source/target) cannot be identified. To achieve this invariance, the neural model minimizes the domain distance in the hypothesis space as, $d_H(D_S, D_T)$, where d_H is the H-divergence.

2. The learned invariant representation should contain enough meaningful information to predict source and (few) target labels accurately. This minimizes the

empirical error $\hat{\epsilon}_{S_i}(h)$ for each source domain S_i, where $\hat{\epsilon}_{S_i}(h)$ is the empirical risk of hypothesis h on the i-th source domain.

Hence, Zhao et al. [249] design a neural model for adversarially learning representation to achieve MSDA. It has three components:

- **Shared feature extractor** G_f: This module maps input samples x to a latent feature space. The feature extractor is shared across all domains and learns a domain-invariant representation.
- **Domain learner** G_d: This module categorizes each sample according to its domain of origin i.e. source/target domain. It is trained using cross-entropy loss $\mathcal{L}_d$ and trained adversarially to ensure that the domain learner cannot distinguish between source and target domains.
- **Label predictor** G_y: This module predicts labels for the source domains. It is trained to minimize the label prediction loss $\mathcal{L}_y$.

The combined training objective is formulated as, $\min_{G_y,G_f} \max_{G_d} \mathcal{L}_y - \lambda \mathcal{L}_d$, where λ balances the trade-off between label prediction and domain invariance.

6.5 MSDA w/ Sparse Variational GP

6.5.1 Sparse Variational Gaussian Processes (SVGPs)

Sparse Variational Gaussian Processes (SVGPs) are a scalable extension of traditional Gaussian Processes (GPs) that address their computational inefficiencies [1, 86, 90].

Traditional GPs suffer from cubic time complexity, making them impractical for large (high dimensional and samples) datasets. SVGPs overcome this by introducing *inducing points*, to summarize the data, thereby, reducing the computational overhead. *Inducing points* are sampled representation of the feature space that summarize the

dataset such that SVGPs can efficiently approximate the complete covariance matrix. Hence, the underlying principle for SVGPs is to approximate the true posterior distribution over functions with a simpler variational distribution and thereby, balance the trade-off between computational efficiency and model accuracy.

Gaussian Processes (GP) are intrinsically suitable for spatiotemporal data as they can handle nonlinear relationships in the data. GPs achieve this by utilizing covariance functions, i.e. kernels, that can be utilized to model both spatial and temporal dependencies as they are able to define how data points are related in the feature space [158]. Hence, GPs can account for smoothness in space, periodicity in time, or non-stationarity, thereby providing a robust framework for modeling a wide range of spatiotemporal processes. This makes GPs useful for applications such as pollution estimation, weather and traffic forecasting, and more. GPs primarily have two characterisitcs making them useful for spatiotemporal modeling:

1. **Uncertainty Modeling:** Gaussian processes can quantify uncertainty in predictions. Since spatiotemporal data is often sparse and unbalanced, GPs provide both point predictions and confidence intervals for those predictions i.e. predictive variances. For tasks such as forecasting and interpolation, it allows to understand the uncertainty present in estimations.

2. **Non-Parametric Nature:** Gaussian processes are non-parametric model, i.e. they do not assume a specific functional form for the underlying data. Instead, GPs adapt to the latent structure of the data. This adaption allows GPs to model a range of complex non-linear relationships and avoid being limited by pre-defined assumptions as is the case for classical parametric models.

The combined strength of GPs with computational scalability is rendered by SVGPs, primarily employed for spatiotemporal data. Hence, SVGPs can model long-range dependencies and complex patterns in space and time by utilizing inducing points

that capture the latent dependecnies. Capturing such dependencies with reduced computational cost is useful for large spatiotemporal data.

6.5.2 MSDA w/ SVGP

Hence, this dissertation can have multiple extensions where the primary focus includes utilizing multiple source datasets for irregular spatiotemporal modeling. While we utilized single-source domain adaptation methodologies for source sample reweighing followed by multi-variate regression; we emphasize the extended domains that utilize multi-source domain adaptation and Gaussian processes, in this chapter. We plan to work on this framework where multiple source domain datasets can generate domain invariant representation using the above MSDA models. These representations will contain spatiotemporal characteristics, that can be successfully modeled using SVGPs for a robust prediction.

6.6 MSDA and Heterogeneous Data

6.6.1 Heterogeneous Data

Heterogeneous data consist of variance such that there can exits similarities between samples of two feature as well as dissimilarity within a feature [78]. These dissimilarities can occur due to data collections methods causing varying distribution, or characteristics of features in the dataset. Within a dataset, heterogeneity is present due to difference in the ranges of numerical values, varying categorical variables, or irregularities in data due to missing values [247, 146]. Across datasets, heterogeneity can occur due to diverse sampling techniques, demographic and environmental disparities, or variations caused by differences in data collection methods [149]. Hence, understanding heterogeneity present within the data allows to improve accuracy of machine learning models as heterogeneous data can cause biased training, unstable

model convergence, and poor generalization of accuracy. For eg., in a distributed learning environment such as federated learning, where models from diverse clients is aggregated into a single model, heterogeneous data causes varying model convergence for the client and the server models [160].

Heterogeneity in data can be computed by various statistical and computational methods. For heterogeneity within a dataset, statistical measures such as standard deviation, variance, or entropy can be used to quantify variance in numerical and categorical attributes (features) [214]. Visualization techniques such as boxplots, scatterplots, and histograms can be used to provide visual descriptions of the variance in the data [211]. For heterogeneity between the datasets, statistical tests like ANOVA for numerical data and chi-square tests for categorical data can be utilizes [214]. Additionally, data mining techniques such as clustering techniques [214], sum-product networks [186], or mixture models [80] can be utilized to extract the underlying patterns in heterogeneous data. While heterogeneous data is challenging to process and analyze, it's variability can be utilized in scenarios that require dataset division based on distribution. We explore such a scenario in the following section.

6.6.2 MSDA w/ Heterogeneous Data

Multi-source domain adaptation (MSDA) techniques can be utilized to take advantage of the variability present in the dataset by generating multiple distinct source domain datasets [255]. This can be achieved by partitioning the data into clusters based on its structural or statistical differences, such as variations in feature distribution, demographic profiles, environmental factors, or sampling methods. Each cluster becomes a source domain containing some subset of the data samples with certain representative variability. This allows localized machine learning models to learn patterns and relations within the cluster and generate improved estimations. Consequently, their learning can then be generalized across clusters by employing MSDA techniques.

The distinct clusters created during MSDA act as source domain datasets and thereby allow easier alignment with the target domain dataset. Hence for applications such as $PM_{2.5}$ prediction, the data from different geographical or climatic regions could be clustered into a set of distinct source datasets where each cluster represents localized variations in such factors as varied pollution sources or weather conditions. This could be effective when adapting to a target domain belonging to a new geographic or climatic region with limited data as the MSDA model can utilize the diversity within the source datasets to bridge the gap between the source and target domains. MSDA techniques such as distribution alignment, or domain-specific feature extraction [69, 52] allows to integrate the knowledge from multiple sources into the target domain and hence can be translated to the task of MSDA with heterogeneous data.

Scalable solutions via Spatial Indexing

6.7 Spatial Indexing

Spatial data (also geospatial data), represents the location and characteristics of objects or phenomena on Earth [39]. It is defined by coordinates such as latitude and longitude, stating the position of an object in two or more dimensions. Additionally, it may also involve additional attributes describing the properties of the objects. *Spatial indexing* is a methodology that allows to manage, query, and retrieve spatial data efficiently from geographic information systems and spatial databases [58, 144, 154, 260]. It achieves this by organizing the spatial objects, such as points, lines, and polygons, within the multidimensional space, to reduce the computational overhead [242]. Hence the space is partitioned into smaller regions or hierarchical structures, allowing techniques to optimize search space traversal for tasks like determining intersections, or nearest neighbors. Spatial indexing can be achieved in multiple ways by using grid indexes [170],

quadtree indexes [114], and R-tree indexes based on the task/domain [82, 12]. For eg., grid indexes split space into uniform grids for simpler datasets, whereas R-tree indexes are suitable for dynamic datasets as they create a hierarchical bounding-box structure for splitting. Recent developments in spatial indexing introduce distributed spatial indexing that is based on big data frameworks such as Hadoop and Spark, that allows large-scale analysis of huge corpora or datasets for applications in urban planning, disaster management, and environmental monitoring [240, 96].

6.7.1 Spatial Indexing for Spatial Transfer Learning

For spatial transfer learning introduced in the previous chapters, spatial indexes can be utilized to find similar geographic locations on a map, thereby allowing to efficiently compare them based on proximity, elevation, land use, and other geospatial attributes [261, 262]. It can also be employed for alternative tasks (non transfer learning) such as sensor placement optimization [116], or data interpolation. For eg., using R-tree indexes, we can instantly locate neighboring $PM_{2.5}$ sensors or identify regions with similar environmental conditions, such as similar forest cover or urban density. These indexes also allow clustering of geographic points into regions with homogeneous characteristics, generating multiple domains (regions) for a multi-source domain adaptation framework [94]. The meteorological and topographical features can also be incorporated with spatial indexing to calculate similarity between locations. Hence, the ability of spatial indexing to efficiently query and compare geographic locations allows for accurate and efficient model training.

Chapter 7

Conclusion

In this dissertation, I introduce a suite of methodologies to solve the complex task of transfer learning for irregular spatiotemporal data. The irregularity in spatiotemporal data is represented by missing temporal points and sparse spatial locations. Given the machine learning modeling for such datasets is already complicated due to existing dependencies (spatial and semantic) as well as the non-i.i.d nature of the data distribution; the existence of irregularities and the translation of the problem to a transfer learning task, pushes the envelope even further.

Hence, my dissertation goal is to design transfer learning models for such complex and irregular spatiotemporal data. I pose and answer three research questions spanning innovation in generalizability, and cross-region transfer with varying feature space. For the first research objective, I design a boosting based transfer learning model that utilizes importance sampling and a balanced weighing approach to consistently outperform baselines. This methodology is called S-TrAdaBoost.R2 and is a successor of a boosting based regression transfer model, TrAdaBoost.R2. This objective provides insight into transfer learning models suitable for regression problems, thereby, laying the groundwork for understanding single source domain adaptation models for regression that intersects with the spatiotemporal estimation task.

For the second research objective, I focus on the use-case of air pollution (especially $PM_{2.5}$) modeling using transfer learning for regions that have a shortage of ground-sensors. This problem is also called as *spatial* transfer learning as the task involves transferring knowledge across regions. In this objective I focus only on regions within a country that share the same feature space. In addition to the feature space, they also share seasonality and hemisphere that impacts the meteorological and topographical variations present in the $PM_{2.5}$ data. I design a novel solution called the *Latent Dependency Factor* (LDF), that is a new feature imputed in both source and target feature space. This feature successfully captures the spatial and semantic dependencies present in the dataset. It is generated using a two-stage autoencoder model. The results show that imputing the LDF feature improves the transfer accuracy by 19.34%.

For the third research objective, I focus on transfer learning across regions with dissimilar feature space. The goal is to apply transfer between two countries with large spatial distance. Often the seasonality between the two countries are also varying as is our case where source region is United States and the target region is Lima, Peru. These countries are highly geographically distant causing complex meteorological and topographical variations. I first perform feature standardization across the datasets and consequently apply the LDF approach with two variations of it – the first consists of seasonal agnosticism and the second consists of seasonal matching. The results show that seasonal matching is effective for short-range (eg., one season) estimation whereas the long-range (eg., complete year) estimation did not require seasonal matching.

The results from these models lay a prospective foundation for transfer learning for irregular spatiotemporal data. I also provide future directions this research can be extended to that involves using multi-source domain adaptation models, gaussian processes, incorporating heterogeneity of the data and scaling via spatial indexing.

Bibliography

[1] Vincent Adam, Stefanos Eleftheriadis, Artem Artemev, Nicolas Durrande, and James Hensman. Doubly sparse variational gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 2874–2884. PMLR, 2020.

[2] Federico Amato, Fabian Guignard, Sylvain Robert, and Mikhail Kanevski. A novel framework for spatio-temporal prediction of environmental data using deep learning. *Scientific reports*, 10(1):22243, 2020.

[3] Hossein Amiri, Shiyang Ruan, Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Zufle. Massive trajectory data based on patterns of life. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–4, 2023.

[4] Chrysovalantis Anastasiou, John Krumm, and Cyrus Shahabi. Time-variant road network-based bridgelets. In *2023 24th IEEE International Conference on Mobile Data Management (MDM)*, pages 265–273. IEEE, 2023.

[5] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. *Advances in neural information processing systems*, 19, 2006.

[6] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[7] GP Ayers, MD Keywood, and JL Gras. Teom vs. manual gravimetric methods for determination of pm2. 5 aerosol mass concentrations. *Atmospheric Environment*, 33(22):3717–3721, 1999.

[8] Hamada S Badr, Benjamin F Zaitchik, and Seth D Guikema. Application of statistical models to the prediction of seasonal rainfall anomalies over the sahel. *Journal of Applied meteorology and climatology*, 53(3):614–636, 2014.

[9] Vigneshkumar Balamurugan, Jia Chen, Adrian Wenzel, and Frank N Keutsch. Spatiotemporal modeling of air pollutant concentrations in germany using machine learning. *Atmospheric Chemistry and Physics*, 23(17):10267–10285, 2023.

[10] Anthony G Barnston, Michael K Tippett, Michelle L L'Heureux, Shuhua Li, and David G DeWitt. Skill of real-time seasonal enso model predictions during 2002–11: Is our capability increasing? *Bulletin of the American Meteorological Society*, 93(5):631–651, 2012.

[11] Md Abul Bashar, Richi Nayak, and Nicolas Suzor. Regularising lstm classifier by transfer learning for detecting misogynistic tweets with small training set. *Knowledge and Information Systems*, pages 1–26, 2020.

[12] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, pages 322–331, 1990.

[13] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[14] Jianzhao Bi, Jessica H Belle, Yujie Wang, Alexei I Lyapustin, Avani Wildani, and Yang Liu. Impacts of snow and cloud covers on satellite-derived pm2. 5 levels. *Remote sensing of environment*, 221:665–674, 2019.

[15] Jianzhao Bi, Avani Wildani, Howard H Chang, and Yang Liu. Incorporating low-cost sensor measurements into high-resolution pm2. 5 modeling at a large spatial scale. *Environmental Science & Technology*, 54(4):2152–2162, 2020.

[16] Jianzhao Bi, K Emma Knowland, Christoph A Keller, and Yang Liu. Combining machine learning and numerical simulation for high-resolution pm2. 5 concentration forecast. *Environmental science & technology*, 56(3):1544–1556, 2022.

[17] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of machine learning research*, 22(2):1–55, 2021.

[18] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 2007.

[19] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

[20] Frederic Branchaud-Charron, Andrew Achkar, and Pierre-Marc Jodoin. Spectral metric for dataset complexity assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3215–3224, 2019.

[21] Michael Brauer, Markus Amann, Rick T Burnett, Aaron Cohen, Frank Dentener, Majid Ezzati, Sarah B Henderson, Michal Krzyzanowski, Randall V Martin,

Rita Van Dingenen, et al. Exposure assessment for estimation of the global burden of disease attributable to outdoor air pollution. *Environmental science & technology*, 46(2):652–660, 2012.

[22] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2002.

[23] Mónica F Bugallo, Luca Martino, and Jukka Corander. Adaptive importance sampling in signal processing. *Digital Signal Processing*, 47:36–49, 2015.

[24] Richard Burnett, Hong Chen, Mieczysław Szyszkowicz, Neal Fann, Bryan Hubbell, C. Arden Pope, Joshua S. Apte, Michael Brauer, Aaron Cohen, Scott Weichenthal, et al. Global estimates of mortality associated with long-term exposure to outdoor fine particulate matter. *Proceedings of the National Academy of Sciences*, 115(38):9592–9597, 2018. doi: 10.1073/pnas.1803222115.

[25] Daniel Camilleri and Tony Prescott. Analysing the limitations of deep learning for developmental robotics. In *conference on Biomimetic and Biohybrid Systems*, pages 86–94. Springer, 2017.

[26] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive transfer learning. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[27] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[28] Sangwon Chae, Joonhyeok Shin, Sungjun Kwon, Sangmok Lee, Sungwon Kang, and Donghyun Lee. Pm10 and pm2. 5 real-time prediction models using an interpolated convolutional neural network. *Scientific Reports*, 11(1):11952, 2021.

[29] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application

to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–26, 2012.

[30] Ling Chen, Yaya Cai, Yifang Ding, Mingqi Lv, Cuili Yuan, and Gencai Chen. Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1076–1087, 2016.

[31] Ling Chen, Jiahui Xu, Binqing Wu, and Jianlong Huang. Group-aware graph neural network for nationwide city air quality forecasting. *ACM Transactions on Knowledge Discovery from Data*, 18(3):1–20, 2023.

[32] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019.

[33] Song Chen. Beijing PM2.5. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C5JS49.

[34] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[35] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[36] Weiwen Cheng, Yanfei Chen, Jordi Zhang, Thomas J Lyons, Jing-Li Pai, and Shih-Heng Chang. Air pollution prediction using ensemble deep learning. *Nature Communications*, 13(1):1–12, 2022.

[37] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54:280–296, 2019.

[38] Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

[39] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

[40] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.

[41] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[42] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[43] Grégoire Mesnil Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, et al. Unsupervised and transfer learning challenge: a deep learning approach. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 97–110, 2012.

[44] Jesse Davis and Pedro Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 26th annual international conference on machine learning*, pages 217–224, 2009.

[45] Oscar Day and Taghi M Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):1–42, 2017.

[46] Department of Energy and Environmental Protection. Deep forecasts unhealthy levels of pm2.5 wednesday for the entire state from canadian wildfire smoke, 2023.

[47] Sagnik Dey, Larry Di Girolamo, Aaron van Donkelaar, SN Tripathi, Tarun Gupta, and Manju Mohan. Variability of outdoor fine particulate (PM2.5) concentration in the indian subcontinent: A remote sensing approach. *Remote sensing of environment*, 127:153–161, 2012.

[48] Thomas G Dietterich and Ryszard S Michalski. A comparative review of selected methods for learning from examples. *Machine learning*, pages 41–81, 1983.

[49] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[50] Harris Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.

[51] Simon Shaolei Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. Hypothesis transfer learning via transformation functions. *arXiv preprint arXiv:1612.01020*, 2016.

[52] Lixin Duan, Ivor W Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th annual international conference on machine learning*, pages 289–296, 2009.

[53] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. *arXiv preprint arXiv:1206.4660*, 2012.

[54] Imad El Haddad, Nicolas Marchand, Henri Wortham, et al. Primary sources of pm 2.5 organic aerosol in an industrial mediterranean city, marseille. *Atmospheric Chemistry and Physics*, 11(5):2039–2058, 2011.

[55] Víctor Elvira, Luca Martino, David Luengo, and Jukka Corander. A gradient adaptive population importance sampler. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4075–4079. IEEE, 2015.

[56] Víctor Elvira, Emilie Chouzenoux, Ömer Deniz Akyildiz, and Luca Martino. Gradient-based adaptive importance samplers. *arXiv preprint arXiv:2210.10785*, 2022.

[57] Ebrahim Eslami, Ali Kais Salman, Yunsoo Choi, Ashique Sayeed, and Yannic Lops. Long short-term memory networks for air pollution forecasting: A comparative study of different approaches. *Environmental Pollution*, 262:114429, 2020.

[58] Martin Ester, Hans-Peter Kriegel, and Jörg Sander. Spatial data mining: A database approach. In *Advances in Spatial Databases: 5th International Symposium, SSD'97 Berlin, Germany, July 15–18, 1997 Proceedings 5*, pages 47–66. Springer, 1997.

[59] Junxiang Fan, Qi Li, Junxiong Hou, Xiao Feng, Hamed Karimian, and Shaofu Lin. A spatiotemporal prediction framework for air pollution based on deep rnn. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:15, 2017.

[60] Xi Fang, Guangcai Gong, Guannan Li, et al. A hybrid deep transfer learning strategy for short term cross-building energy prediction. *Energy*, 215:119208, 2021.

[61] Xiqi Fei, Minh Tri Le, Duy H Thai, Konrad Wessels, and Andreas Züfle. Semi-supervised satellite image segmentation using spatial and temporally informed poisson learning. In *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pages 5642–5645. IEEE, 2023.

[62] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28 (4):594–611, 2006.

[63] Nelson Fernández, Carlos Maldonado, and Carlos Gershenson. Information measures of complexity, emergence, self-organization, homeostasis, and autopoiesis. In *Guided self-organization: Inception*, pages 19–51. Springer, 2014.

[64] Leonardo N Ferreira, Didier A Vega-Oliveros, Moshé Cotacallapa, Manoel F Cardoso, Marcos G Quiles, Liang Zhao, and Elbert EN Macau. Spatiotemporal data analysis with chronological networks. *Nature communications*, 11(1):4036, 2020.

[65] Brian Ferris, Dieter Fox, and Neil D Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[66] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[67] Iat Hang Fong, Tengyue Li, Simon Fong, Raymond K Wong, and Antonio J Tallon-Ballesteros. Predicting concentration levels of air pollutants by transfer learning and recurrent neural network. *Knowledge-Based Systems*, 192:105622, 2020.

[68] Peter Gänssler and Winfried Stute. Empirical processes: a survey of results

for independent and identically distributed random variables. *The Annals of Probability*, 7(2):193–243, 1979.

[69] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291, 2008.

[70] Song Gao. Spatio-temporal analytics for exploring human mobility patterns and urban dynamics in the mobile age. *Spatial Cognition & Computation*, 15(2): 86–114, 2015.

[71] Jochen Garcke and Thomas Vanck. Importance weighted inductive transfer learning for regression. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 466–481. Springer, 2014.

[72] Guannan Geng, Xia Meng, Kebin He, and Yang Liu. Random forest models for pm2. 5 speciation concentrations using misr fractional aods. *Environmental Research Letters*, 15(3):034056, 2020.

[73] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 597–613. Springer, 2016.

[74] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27, 2017.

[75] Stuart K Grange, David C Carslaw, Alastair C Lewis, Eirini Boleti, and Christoph Hueglin. Random forest meteorological normalisation models for

swiss pm 10 trend analysis. *Atmospheric Chemistry and Physics*, 19(24):15117–15129, 2019.

[76] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 379–386, 2015.

[77] Zengda Guan, Ang Li, and Tingshao Zhu. Local regression transfer learning with applications to users' psychological characteristics prediction. *Brain informatics*, 2(3):145–153, 2015.

[78] Badra Souhila Guendouzi, Samir Ouchani, Hiba EL Assaad, and Madeleine EL Zaher. A systematic review of federated learning: Challenges, aggregation methods, and development tools. *Journal of Network and Computer Applications*, page 103714, 2023.

[79] Shrey Gupta, Jianzhao Bi, Yang Liu, and Avani Wildani. Boosting for regression transfer via importance sampling. *International Journal of Data Science and Analytics*, pages 1–12, 2023.

[80] Shrey Gupta, Alireza Karduni, and Emily Wall. Belief decay or persistence? a mixed-method study on belief movement over time. In *Computer Graphics Forum*, volume 42, pages 111–122. Wiley Online Library, 2023.

[81] Shrey Gupta, Yongbee Park, Jianzhao Bi, Suyash Gupta, Andreas Züfle, Avani Wildani, and Yang Liu. Spatial transfer learning for estimating pm 2.5 in data-poor regions. pages 385–400, 2024.

[82] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, 1984.

[83] Eric S Hall, Surender M Kaushik, Robert W Vanderpool, Rachelle M Duvall, Melinda R Beaver, Russell W Long, and Paul A Solomon. Integrating sensor monitoring technology into the current air pollution regulatory support paradigm: Practical considerations. *Am. J. Environ. Eng*, 4(6):147–154, 2014.

[84] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[85] Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572, 2019.

[86] Oliver Hamelijnck, William Wilkinson, Niki Loppi, Arno Solin, and Theodoros Damoulas. Spatio-temporal variational gaussian processes. *Advances in Neural Information Processing Systems*, 34:23621–23633, 2021.

[87] Dongmei Han, Qigang Liu, and Weiguo Fan. A new image classification method using cnn transfer learning and web data augmentation. *Expert Systems with Applications*, 95:43–56, 2018.

[88] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, pages 312–329. Springer, 2020.

[89] Matthew C Hansen, Peter V Potapov, Rebecca Moore, Matt Hancher, Svetlana A Turubanova, Alexandra Tyukavina, David Thau, Stephen V Stehman, Scott J Goetz, Thomas R Loveland, et al. High-resolution global maps of 21st-century forest cover change. *science*, 342(6160):850–853, 2013.

[90] James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

[91] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Learning an invariant hilbert space for domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3845–3854, 2017.

[92] Tin Kam Ho and Mitra Basu. Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 24 (3):289–300, 2002.

[93] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 2737–2746. PMLR, 2019.

[94] Honeycomb Maps. Geospatial indexing: How to get the most out of your data, n.d. URL `https://honeycombmaps.com/blog/geospatial-indexing-how-to-get-the-most-out-of-your-data`.

[95] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2020.

[96] Fei Hu, Chaowei Yang, Yongyao Jiang, Yun Li, Weiwei Song, Daniel Q Duffy, John L Schnase, and Tsengdar Lee. A hierarchical indexing strategy for optimizing apache spark with hdfs to efficiently query big geospatial raster data. *International Journal of Digital Earth*, 13(3):410–428, 2020.

[97] Hailin Hu, MingJian Tang, and Chengcheng Bai. Datsing: Data augmented time series forecasting with adversarial domain adaptation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2061–2064, 2020.

[98] Xuefei Hu, Jessica H Belle, Xia Meng, Avani Wildani, Lance A Waller, Matthew J Strickland, and Yang Liu. Estimating pm2.5 concentrations in the conterminous united states using the random forest approach. *Environmental Science Technology*, 51(12):6936–6944, 2017.

[99] Xuefei Hu, Jessica H Belle, Xia Meng, Avani Wildani, Lance A Waller, Matthew J Strickland, and Yang Liu. Estimating pm2. 5 concentrations in the conterminous united states using the random forest approach. *Environmental science & technology*, 51(12):6936–6944, 2017.

[100] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2006.

[101] Jessica Hwang, Paulo Orenstein, Judah Cohen, Karl Pfeiffer, and Lester Mackey. Improving subseasonal forecasting in the western us with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2325–2335, 2019.

[102] Kazuhiko Ito, Nan Xue, and George Thurston. Spatial variation of pm2. 5 chemical species and source-apportioned mass concentrations in new york city. *Atmospheric Environment*, 38(31):5269–5282, 2004.

[103] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4176–4185. IEEE, 2019.

[104] Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa, Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, and Vidya N Murali. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 772–773, 2020.

[105] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pages 10280–10297. PMLR, 2022.

[106] Mingxuan Jing, Xiaojian Ma, Wenbing Huang, Fuchun Sun, and Huaping Liu. Task transfer by preference-based cost learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2471–2478, 2019.

[107] Zahra Karevan and Johan Suykens. Spatio-temporal feature selection for black-box weather forecasting. In *Proc. of the 24th european symposium on artificial neural networks, computational intelligence and machine learning*, pages 611–616, 2016.

[108] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29 (10):2318–2331, 2017.

[109] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2018.

[110] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[111] Jules Kerckhoffs, Gerard Hoek, Luützen Portengen, Bert Brunekreef, and Roel CH Vermeulen. Performance of prediction algorithms for modeling out-

door air pollution spatial surfaces. *Environmental science & technology*, 53(3): 1413–1421, 2019.

[112] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[113] Patrick L Kinney, Maneesha Aggarwal, Mary E Northridge, Nicole A Janssen, and Peggy Shepard. Airborne concentrations of pm (2.5) and diesel exhaust particles on harlem sidewalks: a community-based pilot study. *Environmental health perspectives*, 108(3):213–218, 2000.

[114] Ravi Kanth V Kothuri, Siva Ravada, and Daniel Abugov. Quadtree and r-tree indexes in oracle spatial: a comparison using gis data. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 546–557, 2002.

[115] Moritz UG Kraemer, Robert C Reiner, Oliver J Brady, Jane P Messina, Marius Gilbert, David M Pigott, Dingdong Yi, Kimberly Johnson, Lucas Earl, Laurie B Marczak, et al. Past and future spread of the arbovirus vectors aedes aegypti and aedes albopictus. *Nature microbiology*, 4(5):854–863, 2019.

[116] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

[117] John Krumm. Maximum entropy bridgelets for trajectory completion. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–8, 2022.

[118] Arun Kumar, Jeffrey Naughton, Jignesh M Patel, and Xiaojin Zhu. To join or not to join? thinking twice about joins before feature selection. In *Proceedings of the 2016 International Conference on Management of Data*, pages 19–34, 2016.

[119] Philip J Landrigan, Richard Fuller, Nereus JR Acosta, Olusoji Adeyi, Robert Arnold, Abdoulaye Balde Baldé, Roberto Bertollini, Stephan Bose-O'Reilly, Jo Ivey Boufford, Patrick N Breysse, et al. The lancet commission on pollution and health. *The Lancet*, 391(10119):462–512, 2018. doi: 10.1016/S0140-6736(17)32345-0.

[120] D J Lary et al. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10, 2016.

[121] Neil D Lawrence and John C Platt. Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on Machine learning*, page 65, 2004.

[122] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[123] Jos Lelieveld, John S Evans, Mohammed Fnais, Despina Giannadaki, and Andrea Pozzer. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature*, 525(7569):367–371, 2015.

[124] Lixin Li, Xiaolu Zhou, Marc Kalo, and Reinhard Piltner. Spatiotemporal interpolation methods for the application of estimating population exposure to fine particulate matter in the contiguous us and a real-time web application. volume 13, page 749. MDPI, 2016.

[125] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul MB Vitányi. The similarity metric. *IEEE transactions on Information Theory*, 50(12):3250–3264, 2004.

[126] Tongwen Li, Huanfeng Shen, Qiangqiang Yuan, Xuechen Zhang, and Liangpei Zhang. Estimating ground-level pm2.5 by fusing satellite and station observations: A geo-intelligent deep learning approach. *Geophysical Research Letters*, 44(23):11–985, 2017.

[127] Xintong Li and Xiaodong Zhang. Predicting ground-level pm2. 5 concentrations in the beijing-tianjin-hebei region: A hybrid remote sensing and machine learning approach. *Environmental pollution*, 249:735–749, 2019.

[128] Zijian Li, Ruichu Cai, Jiawei Chen, Yuguan Yan, Wei Chen, Keli Zhang, and Junjian Ye. Time-series domain adaptation via sparse associative structure alignment: Learning invariance and variance. *Neural Networks*, 180, 2024.

[129] Y Liang et al. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3428–3434, 2018.

[130] Umesh Kumar Lilhore, Agbotiname Lucky Imoize, Chun-Ta Li, Sarita Simaiya, Subhendu Kumar Pani, Nitin Goyal, Arun Kumar, and Cheng-Chi Lee. Design and implementation of an ml and iot based adaptive traffic-management system for smart cities. *Sensors*, 22(8):2908, 2022.

[131] Cong Liu, Renjie Chen, Francesco Sera, Ana M Vicedo-Cabrera, Yuming Guo, Shilu Tong, Micheline SZS Coelho, Paulo HN Saldiva, Eric Lavigne, Patricia Matus, et al. Ambient particulate air pollution and daily mortality in 652 cities. *New England Journal of Medicine*, 381(8):705–715, 2019.

[132] Heng Liu, Qingyong Li, Dexiang Yu, and Yu Gu. A novel air quality early-warning system based on artificial intelligence. *Science of The Total Environment*, 658:656–667, 2019.

[133] Jiabin Liu, Chengliang Chai, Yuyu Luo, Yin Lou, Jianhua Feng, and Nan Tang. Feature augmentation with reinforcement learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3360–3372. IEEE, 2022.

[134] Siyuan Liu, Ce Liu, Qiong Luo, Lionel M Ni, and Ramayya Krishnan. Calibrating large scale vehicle trajectory data. In *2012 IEEE 13th International Conference on Mobile Data Management*, pages 222–231. IEEE, 2012.

[135] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, 2019.

[136] Marco Loog. Nearest neighbor-based importance weighting. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2012.

[137] Ana C Lorena, Aron I Maciel, Péricles BC de Miranda, Ivan G Costa, and Ricardo BC Prudêncio. Data complexity meta-features for regression problems. *Machine Learning*, 107(1):209–246, 2018.

[138] Mingqi Lv, Yifan Li, Ling Chen, and Tieming Chen. Air quality estimation by exploiting terrain features and multi-view transfer semi-supervised regression. *Information Sciences*, 483:82–95, 2019.

[139] Jun Ma, Jack CP Cheng, Changqing Lin, Yi Tan, and Jingcheng Zhang. Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. *Atmospheric Environment*, 214: 116885, 2019.

[140] Jun Ma, Zheng Li, Jack CP Cheng, Yuexiong Ding, Changqing Lin, and Zherui Xu. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Science of The Total Environment*, 705:135771, 2020.

[141] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019.

[142] Aron I Maciel, Ivan G Costa, and Ana C Lorena. Measuring the complexity of regression problems. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1450–1457. IEEE, 2016.

[143] MM Mahmud and Sylvian R Ray. Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations. Technical report, 2007.

[144] Yannis Manolopoulos, Yannis Theodoridis, and Vassilis J. Tsotras. Spatial indexing techniques. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*. Springer, New York, NY, 2014. doi: 10.1007/978-1-4899-7993-3_355-3. URL `https://doi.org/10.1007/978-1-4899-7993-3_355-3`.

[145] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21, 2008.

[146] Ivan Marisca, Cesare Alippi, and Filippo Maria Bianchi. Graph-based forecasting with missing data through spatiotemporal downsampling. In *Forty-first International Conference on Machine Learning*.

[147] Shahir Masri, Eric Garshick, Brent A Coull, and Petros Koutrakis. A novel calibration approach using satellite and visibility observations to estimate fine particulate matter exposures in southwest asia and afghanistan. *Journal of the Air & Waste Management Association*, 67(1):86–95, 2017.

[148] Suyu Mei and Hao Zhu. Adaboost based multi-instance transfer learning for

predicting proteome-wide interactions between salmonella and human proteins. *PloS one*, 9(10):e110488, 2014.

[149] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

[150] Xia Meng, Cong Liu, Lina Zhang, Weidong Wang, Jennifer Stowell, Haidong Kan, and Yang Liu. Estimating pm2. 5 concentrations in northeastern china with full spatiotemporal coverage, 2005–2016. *Remote sensing of environment*, 253:112203, 2021.

[151] Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. Mapping and revising markov logic networks for transfer learning. In *Aaai*, volume 7, pages 608–614, 2007.

[152] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Taylor Anderson, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, et al. Mobility data science: Perspectives and challenges. *ACM Transactions on Spatial Algorithms and Systems*, 2024.

[153] Lidia Morawska, Phong K Thai, Xiaoting Liu, Akwasi Asumadu-Sakyi, Godwin Ayoko, Alena Bartonova, Andrea Bedini, Fahe Chai, Bryce Christensen, Matthew Dunbabin, et al. Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone? *Environment international*, 116:286–299, 2018.

[154] Jussi Myllymaki and James Kaufman. Dynamark: A benchmark for dynamic spatial indexing. In *Mobile Data Management: 4th International Conference,*

MDM 2003 Melbourne, Australia, January 21–24, 2003 Proceedings 4, pages 92–105. Springer, 2003.

[155] Atsushi Nara. Space-time gis and its evolution. In Bo Huang, editor, *Comprehensive Geographic Information Systems*, pages 287–302. Elsevier, 2018. ISBN 9780128047934. doi: 10.1016/B978-0-12-409548-9.09626-3.

[156] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[157] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

[158] Kien Nguyen, John Krumm, and Cyrus Shahabi. Gaussian process for trajectories. In *Spatial Gems, Volume 2*, pages 37–48. 2023.

[159] Jianjun Ni, Yan Chen, Yu Gu, Xiaolong Fang, and Pengfei Shi. An improved hybrid transfer learning-based deep learning model for pm2. 5 concentration prediction. *Applied Sciences*, 12(7):3597, 2022.

[160] Maxence Noble, Aurélien Bellet, and Aymeric Dieuleveut. Differentially private federated learning on heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 10110–10145. PMLR, 2022.

[161] David Obst, Badih Ghattas, Jairo Cugliari, Georges Oppenheim, Sandra Claudel, and Yannig Goude. Transfer learning for linear regression: A statistical test of gain. *arXiv preprint arXiv:2102.09504*, 2021.

[162] Narendra Ojha, Imran Girach, Kiran Sharma, Amit Sharma, Narendra Singh,

and Sachin S Gunthe. Exploring the potential of machine learning for simulations of urban ozone variability. *Scientific reports*, 11(1):22513, 2021.

[163] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, and Ian J Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *arXiv preprint arXiv:1804.09170*, 2018.

[164] Jeffrey Junfeng Pan, Qiang Yang, Hong Chang, and Dit-Yan Yeung. A manifold regularization approach to calibration reduction for sensor-network based tracking. In *AAAI*, volume 6, pages 988–993, 2006.

[165] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[166] Sinno Jialin Pan, Dou Shen, Qiang Yang, and James T Kwok. Transferring localization models across space. In *AAAI*, pages 1383–1388, 2008.

[167] Weike Pan, Evan Xiang, Nathan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[168] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1720–1730, 2019.

[169] David Pardoe and Peter Stone. Boosting for regression transfer. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 863–870, 2010.

[170] Kwangjin Park. Location-based grid-index for spatial query processing. *Expert systems with applications*, 41(4):1294–1300, 2014.

[171] Yongbee Park, Byungjoon Kwon, Juyeon Heo, Xuefei Hu, Yang Liu, and Taesup Moon. Estimating pm2. 5 concentration of the conterminous united states via interpretable convolutional neural networks. *Environmental Pollution*, 256: 113395, 2020.

[172] C Arden Pope III and Douglas W Dockery. Health effects of fine particulate air pollution: lines that connect. *Journal of the air & waste management association*, 56(6):709–742, 2006.

[173] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2016.

[174] Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li, and Zhongfei Zhang. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2285–2297, 2018.

[175] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021.

[176] Minghui Qiu, Peilin Zhao, Ke Zhang, Jun Huang, Xing Shi, Xiaoguang Wang, and Wei Chu. A short-term rainfall prediction model using multi-task convolutional neural networks. In *2017 IEEE international conference on data mining (ICDM)*, pages 395–404. IEEE, 2017.

[177] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather

events. In *Advances in Neural Information Processing Systems*, pages 3402–3413, 2017.

[178] Ramya Ramakrishnan and Julie Shah. Towards interpretable explanations for transfer learning in sequential tasks. *AAAI 2016 Spring Symposium*, 2016.

[179] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: A benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12 (11):e2020MS002203, 2020.

[180] M Mazhar Rathore, Awais Ahmad, Anand Paul, and Seungmin Rho. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, 101:63–80, 2016.

[181] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and F Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

[182] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

[183] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

[184] Feras Saad, Jacob Burnim, Colin Carroll, Brian Patton, Urs Köster, Rif A. Saurous, and Matthew Hoffman. Scalable spatiotemporal prediction with bayesian neural fields. *Nature Communications*, 15(1):7942, 2024.

[185] Syed Moshfeq Salaken, Abbas Khosravi, Thanh Nguyen, and Saeid Nahavandi. Seeded transfer learning for regression problems with deep learning. *Expert Systems with Applications*, 115:565–577, 2019.

[186] Raquel Sánchez-Cauce, Iago París, and Francisco Javier Díez. Sum-product networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3821–3839, 2021.

[187] Makiko Sato, James E Hansen, M Patrick McCormick, and James B Pollack. Stratospheric aerosol optical depths, 1850–1990. *Journal of Geophysical Research: Atmospheres*, 98(D12):22987–22994, 1993.

[188] Robert E Schapire et al. A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401–1406. Citeseer, 1999.

[189] Ingmar Schuster. Gradient importance sampling. *arXiv preprint arXiv:1507.05781*, 2015.

[190] Gavin Shaddick, Matthew L Thomas, Amelia Green, Michael Brauer, Aaron van Donkelaar, Rick Burnett, Howard H Chang, Aaron Cohen, Rita Van Dingenen, Carlos Dora, et al. Data integration model for air quality: a hierarchical approach to the global estimation of exposures to ambient air pollution. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(1):231–253, 2018.

[191] Shubham Sharma, Mina Chandra, and Sri Harsha Kota. Health effects associated with pm 2.5: A systematic review. *Current Pollution Reports*, 6:345–367, 2020.

[192] C Shen. A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, 54(11):8558–8593, 2018.

[193] X Shi et al. Deep learning for precipitation nowcasting: A benchmark and a new

model. In *Advances in Neural Information Processing Systems*, pages 5617–5627, 2017.

[194] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[195] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[196] Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. A two-stage weighting framework for multi-source domain adaptation. *Advances in neural information processing systems*, 24, 2011.

[197] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[198] Yijun Sun, Sinisa Todorovic, and Jian Li. Reducing the overfitting of adaboost by controlling its data distribution skewness. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(07):1093–1116, 2006.

[199] Samarth Swarup and Sylvian R Ray. Cross-domain knowledge transfer using structured representations. In *Aaai*, volume 6, pages 506–511, 2006.

[200] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[201] Dongjie Tang, Xiaohan Yang, and Xuesong Wang. Improving the transferability of the crash prediction model using the tradaboost. r2 algorithm. *Accident Analysis & Prevention*, 141:105551, 2020.

[202] Vilma Tapia, Kyle Steenland, Bryan Vu, Yang Liu, Vanessa Vásquez, and Gustavo F Gonzales. Pm 2.5 exposure on daily cardio-respiratory mortality in lima, peru, from 2010 to 2016. *Environmental Health*, 19:1–7, 2020.

[203] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[204] Aaron van Donkelaar, Randall V Martin, Michael Brauer, and Brian L Boys. Use of satellite observations for long-term exposure assessment of global concentrations of fine particulate matter. *Environmental Health Perspectives*, 123(2): 135–143, 2015.

[205] Irina Ren Vasiliev. Visualization of spatial dependence: an elementary view of spatial autocorrelation. In *Practical handbook of spatial statistics*, pages 17–30. CRC Press, 2020.

[206] A Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[207] Amir Pouran Ben Veyseh, Minh Van Nguyen, Bonan Min, and Thien Huu Nguyen. Augmenting open-domain event detection with synthetic data from gpt-2. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pages 644–660. Springer, 2021.

[208] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[209] Antonín Vobeckỳ, David Hurych, Michal Uřičář, Patrick Pérez, and Josef Sivic. Artificial dummies for urban dataset augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2692–2700, 2021.

[210] Bryan N Vu, Odón Sánchez, Jianzhao Bi, Qingyang Xiao, Nadia N Hansel, William Checkley, Gustavo F Gonzales, Kyle Steenland, and Yang Liu. Developing an advanced pm2. 5 exposure model in lima, peru. *Remote sensing*, 11(6): 641, 2019.

[211] Emily Wall, Subhajit Das, Ravish Chawla, Bharath Kalidindi, Eli T Brown, and Alex Endert. Podium: Ranking data using mixed-initiative visual analytics. *IEEE transactions on visualization and computer graphics*, 24(1):288–297, 2017.

[212] Boyu Wang, Jorge A Mendez, Ming Bo Cai, and Eric Eaton. Transfer learning via minimizing the performance gap between domains. *Advances in Neural Information Processing Systems*, 2019.

[213] Jingyang Wang, Jiazheng Li, Xiaoxiao Wang, Jue Wang, and Min Huang. Air quality prediction using ct-lstm. *Neural Computing and Applications*, 33: 4779–4792, 2021.

[214] Lidong Wang. Heterogeneous data and big data analytics. *Automatic Control and Information Sciences*, 3(1):8–15, 2017.

[215] Tianyang Wang, Jun Huan, and Michelle Zhu. Instance-based deep transfer learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 367–375. IEEE, 2019.

[216] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

[217] Yutong Wang, Yu Zhao, Yiming Liu, Yueqi Jiang, Bo Zheng, Jia Xing, Yang Liu, Shuai Wang, and Chris P Nielsen. Sustained emission reductions have restrained the ozone pollution over china. *Nature Geoscience*, 16(11):967–974, 2023.

[218] Zhecheng Wang and Xiaoyue Ye. Urban2vec: Incorporating street view imagery and pois for multi-modal urban neighborhood embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pages 985–995, 2020.

[219] Zhen-bo Wang and Chuang-lin Fang. Spatial-temporal characteristics and determinants of pm2. 5 in the bohai rim urban agglomeration. *Chemosphere*, 148:148–162, 2016.

[220] Pengfei Wei, Ramon Sagarna, Yiping Ke, and Yew Soon Ong. Uncluttered domain sub-similarity modeling for transfer regression. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1314–1319. IEEE, 2018.

[221] Yong Wei, Yongqiang Wang, Qian Di, Alexandra Chouldechova, Petros Koutrakis, Francesca Dominici, Joel D Schwartz, and Antonella Zanobetti. A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine. *Science of The Total Environment*, 745:140975, 2020.

[222] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[223] Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1768–1778, 2020.

[224] Xiao Wu, Rachel C Nethery, M Benjamin Sabath, Danielle Braun, and Francesca Dominici. Air pollution and covid-19 mortality in the united states: Strengths and limitations of an ecological regression analysis. *Science advances*, 6(45): eabd4049, 2020.

[225] Yuankai Wu, Huaxiu Wang, Xian Zhang, and Yaguang Zhu. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4478–4485, 2021.

[226] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1907–1913, 2019.

[227] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[228] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591, 2017.

[229] Qingyang Xiao, Yafeng Wang, Howard H Chang, Xia Meng, Guannan Geng, Alexei Lyapustin, and Yang Liu. Full-coverage high-resolution daily pm2.5 estimation using maiac aod in the yangtze river delta of china. *Remote Sensing of Environment*, 221:373–383, 2018.

[230] Rongbin Xu, Tingting Ye, Xu Yue, Zhengyu Yang, Wenhua Yu, Yiwen Zhang, Michelle L Bell, Lidia Morawska, Pei Yu, Yuxi Zhang, et al. Global population exposure to landscape fire air pollution from 2000 to 2019. *Nature*, 621(7979): 521–529, 2023.

[231] Shenyi Xu, Wei Li, Yuhan Zhu, and Aiting Xu. A novel hybrid model for six main pollutant concentrations forecasting based on improved lstm neural networks. *Scientific Reports*, 12(1):14434, 2022.

[232] Xuemiao Xu, Hai He, Huaidong Zhang, Yangyang Xu, and Shengfeng He. Unsupervised domain adaptation via importance sampling. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4688–4699, 2019.

[233] Kalpit Yadav, Vipul Arora, Mohit Kumar, Sachchida Nand Tripathi, Vidyanand Motiram Motghare, and Karansingh A Rajput. Few-shot calibration of low-cost air pollution (pm _{2.5}) sensors using meta learning. *IEEE Sensors Letters*, 6(5):1–4, 2022.

[234] Bozhi Yao, Guang Ling, Feng Liu, and Ming-Feng Ge. Multi-source variational mode transfer learning for enhanced pm2. 5 concentration forecasting at data-limited monitoring stations. *Expert Systems with Applications*, 238:121714, 2024.

[235] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*, pages 2181–2191, 2019.

[236] Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[237] Zhiyu Yao, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Unsupervised transfer learning for spatiotemporal predictive networks. In *International Conference on Machine Learning*, pages 10778–10788. PMLR, 2020.

[238] Maayan Yitshak-Sade, Jennifer F Bobb, Joel D Schwartz, Itai Kloog, and Antonella Zanobetti. The association between short and long-term exposure

to pm2. 5 and temperature and hospital admissions in new england and the synergistic effect of the short-term exposures. *Science of the total environment*, 639:868–875, 2018.

[239] Chaohui Yu, Jindong Wang, Yiqiang Chen, and Meiyu Huang. Transfer learning with dynamic adversarial adaptation network. In *2019 IEEE international conference on data mining (ICDM)*, pages 778–786. IEEE, 2019.

[240] Jia Yu, Jinxuan Wu, and Mohamed Sarwat. Geospark: A cluster computing framework for processing large-scale spatial data. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, pages 1–4, 2015.

[241] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[242] Fatemeh Zardbani, Nikos Mamoulis, Stratos Idreos, and Panagiotis Karras. Adaptive indexing of objects with spatial extent. *Proceedings of the VLDB Endowment*, 16(9):2248–2260, 2023.

[243] Scott L Zeger, Duncan Thomas, Francesca Dominici, et al. Exposure measurement error in time-series studies of air pollution: concepts and consequences. *Environmental health perspectives*, 108(5):419–426, 2000.

[244] Yu Zhan, Yongming Luo, Xiang Deng, Michael L Grieneisen, Minghua Zhang, and Baofeng Di. Spatiotemporal prediction of daily ambient ozone levels across china using random forest for human exposure assessment. *Environmental Pollution*, 233:464–473, 2018.

[245] Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780, 2019.

[246] Lefei Zhang and Liangpei Zhang. Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):270–294, 2022.

[247] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678, 2021.

[248] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

[249] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[250] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9. PMLR, 2015.

[251] Evgenii Zheltonozhskii, Chaim Baskin, Alex M Bronstein, and Avi Mendelson. Self-supervised learning for large-scale unsupervised image clustering. *arXiv preprint arXiv:2008.10312*, 2020.

[252] Vincent Wenchen Zheng, Evan Wei Xiang, Qiang Yang, and Dou Shen. Transferring localization models over time. In *AAAI*, volume 2008, pages 1421–1426, 2008.

[253] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444, 2013.

[254] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–55, 2014.

[255] Joey Tianyi Zhou, Ivor W Tsang, Sinno Jialin Pan, and Mingkui Tan. Multi-class heterogeneous domain adaptation. *Journal of Machine Learning Research*, 20 (57):1–31, 2019.

[256] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Unsupervised learning from video with deep neural embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9563–9572, 2019.

[257] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[258] Andreas Zuefle, Konrad Wessels, and Dieter Pfoser. Mining high resolution earth observation data cubes. In *Proceedings of the 17th International Symposium on Spatial and Temporal Databases*, pages 152–156, 2021.

[259] Andreas Züfle. *Similarity search and mining in uncertain spatial and spatio-temporal databases*. PhD thesis, lmu, 2013.

[260] Andreas Züfle. Uncertain spatial data management: An overview. *Handbook of Big Geospatial Data*, pages 355–397, 2020.

[261] Andreas Züfle, Goce Trajcevski, Dieter Pfoser, Matthias Renz, Matthew T Rice, Timothy Leslie, Paul Delamater, and Tobias Emrich. Handling uncertainty in geo-spatial data. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1467–1470. IEEE, 2017.

[262] Andreas Züfle, Goce Trajcevski, Dieter Pfoser, and Joon-Seok Kim. Managing uncertainty in evolving geo-spatial data. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 5–8. IEEE, 2020.

[263] Andreas Züfle, Carola Wenk, Dieter Pfoser, Andrew Crooks, Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, and Hyunjee Jin. Urban life: a model of people and places. *Computational and Mathematical Organization Theory*, 29(1):20–51, 2023.

[264] Hua Zuo, Guangquan Zhang, Witold Pedrycz, Vahid Behbood, and Jie Lu. Fuzzy regression transfer learning in takagi–sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 25(6):1795–1807, 2016.